Aptima®
Human - Centered Engineering

Massachusetts Headquarters : 781-935-3966
Washington DC Office : 202-842-1548

AP-R-1280

# NEMESIS: NEtwork Modeling Environment for Structural Intervention Strategies

Final Technical Report
Year 2
Deliverable 000203

March 18, 2005

Webb Stacy, William Salter, Daniel Serfaty, Jean MacMillan, Kari Chopra, Georgiy Levchuk, Tim Burt, Dan Lecours, Gilbert Mizrahi, John Colonna-Romano, Matt Rantz
*Aptima, Inc.*

Krishna Pattipati, Peter Willett, Haiying Tu, Satnam Singh, Jeff Allanach
*University of Connecticut*

Kathleen Carley , Jeff Reminga
*Carnegie Mellon University*

**Submitted to:**
Tandi Paugh
Situational Awareness Branch (AFRL/IFED)
525 Brooks Rd
Rome, NY 13441
(315) 330-2910 [DSN 587]
Tandi.Paugh@rl.af.mil



Contract No: MDA972-03-C-0039
Contractor Name: Aptima, Inc.
Contractor Address: 12 Gill St., Suite 1400, Woburn, MA 01801


"Sponsored by Defense Advanced Research Projects Agency
Information Awareness Office (IAO)
Program: NEMESIS: Network Modeling Environment for Structural Intervention Strategies
ARPA OrderNo. P427/00/01, Program Code: 2E20
Issued by DARPA/CMO under Contract No. MDA972-03-C-0039"

Aptima®
Human - Centered Engineering

Massachusetts Headquarters : 781-935-3966
Washington DC Office : 202-842-1548

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



| 1. AGENCY USE ONLY *(Leave blank)* | 2. REPORT DATE<br>19 March 2005 | 3. REPORT TYPE AND DATES COVERED<br>3/19/03 – 3/19/05 |
|---|---|---|

4. TITLE AND SUBTITLE
NEMESIS: Network Modeling Environment for Structural Intervention strategies
Final Technical Report

5. FUNDING NUMBERS
MDA972-03-C-0039

6. AUTHORS

See cover page of report

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Aptima, Inc.
12 Gill Street, Suite 1400
Woburn, MA 01801

8. PERFORMING ORGANIZATION REPORT NUMBER
AP-R-1280

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFRL/IFKF
26 Electronics Parkway
Rome, NY 13441-4514

10. SPONSORING/MONITORING AGENCY
AFRL/IFKF

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unlimited Rights

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

NEtwork Modeling Environment for Structural Intervention Strategies (NEMESIS) is a collaboration and integration environment for state-of-the-art network organizational analysis tools. It provides a mechanism for versioning of collaborative artifacts such as models and reports that is coordinated with analysts' worflow (in the form of local task lists), for maximum convenience and minimum obtrusiveness. The NEMESIS integration architecture is bus-oriented and is centered around an XML Schema based language named Organizational Description Language (ODL), which provides a basis for tools to communicate but also provides a means to accommodate the unique data representation needs of specific tools. One network analytical tool, Adaptive Safety and Monitoring (ASAM), matches transactions against a pattern library and manages the uncertainties of the association. ASAM was extended in the second year of NEMESIS to include multiple model tracking, multiple hypothesis testing, and feature-aided tracking. The result was evaluated by SMEs, and their feedback was incorporated. Another tool, Organizational Risk Analysis (ORA), uses an extended dynamic social network representation to determine organizational strengths and vulnerabilities. ORA was evaluated by SMEs, and their feedback was incorporated. A TIE with Cycorp allowed knowledge from their Terrorism Knowledge Base to be translated to ODL so that the NEMESIS integration environment could feed the result to ORA. The efforts of the second year built on the positive foundation of the first year, along the path of fuller system development and measurement in subsequent years

14. SUBJECT TERMS
Collaboration, Integration, XML Schema, Organizational Analysis, Hidden Markov Models, Dynamic Bayesian Networks, Social Network Analysis, Dynamic Network Analysis

15. NUMBER OF PAGES
117

16. PRICE CODE

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | |

# Table of Contents

List of Figures ..... iii
List of Tables ..... v
Executive Summary ..... vi
Introduction ..... 1
Technical Objectives ..... 1
Task 1. A Functional Transactional Network Modeling Tool: Adaptive Safety and Monitoring (ASAM) ..... 2
Introduction ..... 2
Modeling of a Terrorist Event ..... 3
Advanced Methods for Tracking Terrorist Activities ..... 7
Prototype Platform Participation ..... 10
ASAM Website and Execution ..... 16
Plans for Year 3 ..... 20
Task 2. A Dynamic Social Network Modeling Tool: Organizational Risk Analysis (ORA) ..... 21
Introduction ..... 21
Year 2 Activities ..... 23
Task 3. An Integration Environment for Network Organizational Modeling Tools ..... 24
Architecture ..... 25
Repository ..... 27
Organizational Description Language (ODL) ..... 27
TIE with Cycorp ..... 36
Task 4. Support for Collaboration on Network Organizational Models ..... 43
Collaborative Versioning ..... 43
Collaborative Versionoing and Workflow ..... 47
Installation ..... 49
Initial Creation of CVW Directory ..... 49
Task Lists ..... 50
Collaborative Versioning in CVW ..... 51
Branching and Tagging ..... 53
Adaptation to Counterterrorism Analysts ..... 53

The Future ..... 54
Task 5. Performance Measurement ..... 54
Summary ..... 57
References ..... 58

# List of Figures

Figure 1. Abridged version of Greece Olympics model ..... 4
Figure 2. Markov Chain of Truck Bombing Model ..... 7
Figure 3. MHT for two HMMs ..... 9
Figure 4. Transaction and Features ..... 10
Figure 5. ASAM Launcher Interface ..... 10
Figure 6. Bayesian Network Structure in TEAMS ..... 11
Figure 7. New Model Containing Two Modules ..... 12
Figure 8. BN Node Properties ..... 13
Figure 9. Markov Chain of the HMM Model ..... 14
Figure 10. Prior Probabilities for Each State ..... 14
Figure 11. Transition Probabilities for the HMM States ..... 15
Figure 12. HMM Transaction Editor ..... 16
Figure 13. The ASAM website ..... 17
Figure 14. Real time local evidence ..... 17
Figure 15. Detected transactions. ..... 18
Figure 16. Bayesian inference ..... 18
Figure 17. Editing transactions and associated features ..... 19
Figure 18. Input a new transaction ..... 20
Figure 19. Integration approaches. (a) Pairwise integration; (b) Bus-oriented integration. ..... 25
Figure 20. The NEMESIS integration environment ..... 27
Figure 21. Sample basic ODL. ..... 29
Figure 22. Expansion of Figure 1 that uses Binding elements. ..... 32
Figure 23. Ordinary and modular definitions of the Agent element ..... 33
Figure 24. Component languages of AsamML. ..... 36
Figure 25. Overall meta-network sample visualization data ..... 41
Figure 26. Sphere of influence around Atta and bin Laden ..... 42
Figure 27. Sample document dependencies. ..... 44
Figure 28. Example version management for collaborative versioning. ..... 44
Figure 29. Process and Workflow Spectra ..... 48
Figure 30. Task List Creation and Edit Screen. ..... 50

Figure 31. Windows Explorer context menu for a file under CVW control. ..... 51
Figure 32. A conflict situation. ..... 52
Figure 33. Resolving the conflict. ..... 53
Figure 34. Aptima's approach to team measurement. ..... 55
Figure 35. The three pillars of team performance measurement. ..... 56

# List of Tables

Table 1. Transactions for the Truck Bombing HMM ..... 5
Table 2. Organizational meta-matrix. ..... 22
Table 3. Selected organizational analysis tools from CASOS. ..... 22
Table 4. Namespace considerations for core+extensions architecture for ODL. ..... 34
Table 5. Namespace options for hybrid solution to core+extensions architecture for ODL. ..... 34
Table 6. Comparison of version control requirements for software development and for collaboration. ..... 46

# Executive Summary

Terrorist organizations like al Qaeda operate as *networks* of shadowy, dynamic, globally distributed, stateless individuals and groups, with few fixed assets or addresses, but with sufficient connectivity to achieve highly destructive coordinated actions. Recent advances in network analytic technology show promise for exploiting that connectivity to understand and monitor the hostile networks and to formulate and evaluate the effectiveness of strategies to disable or destroy them.

NEMESIS is aimed at leveraging those technologies to provide an integrated, extensible network modeling environment that can be used for collaboration among analysts. There are three main components to this environment:

- **A collaboration platform.** At one point, Groove Virtual Office from Groove Networks, Inc. had been selected as a collaboration tool; however, requirements of the program have changed, and NEMESIS is now beginning to understand what it will mean to integrate with Vignette Collaboration Server. Beyond that, however, there are special requirements imposed by the need to manage models, documents, and other artifacts of collaboration as analysts' understanding of a hostile network evolve.
- **An integration platform.** The new network analytic technologies display a surprisingly diverse set of approaches, assumptions, and world views, yet each can provide valuable tools that amplify human cognition and at the same time automate what can be automated. It stands to reason that an environment that integrates this diversity to provide multiple views of the same hostile network will also be valuable, especially if it is architected so that new network analytic tools are easily added as they become available.
- **The network analytic tools themselves**. NEMESIS initially focuses on two tools. One uses a transactional network modeling approach to monitor filtered transactions to detect potential matches against known hostile patterns; the other uses a dynamic social network analysis approach that analyzes organizations and their context to detect risks and vulnerabilities.

In addition to these components, it is important to be able to assess the effectiveness of the environment on an ongoing basis, both to provide a means to evaluate it and to provide a means to improve it. For this reason, NEMESIS will involve an ongoing measurement effort.

In furtherance of this mission, five major tasks are involved in each iteration of NEMESIS:

1. **Develop and adapt a functional transactional network modeling tool.** The Adaptive Safety Analysis and Monitoring (ASAM) system is a hybrid model-based software tool for assisting US intelligence analysts to identify terrorist threats, to predict possible evolution of the terrorist activities, and to suggest strategies for countering terrorism. The ASAM system provides a distributed processing structure for gathering, sharing, understanding, and using information to assess and predict terrorist network states. In combination with counter-terrorist network models, it can also suggest feasible actions to inhibit potential terrorist threats. ASAM adopts a hybrid modeling approach: Hidden Markov Models (HMMs) to detect and provide soft evidence on the states of terrorist network nodes based on partial and imperfect observations, and Bayesian networks (BNs) to integrate soft evidence from HMMs. The second year of work elaborated and extended this work in several ways: ASAM can now track multiple targets, test

multiple hypotheses about the state of potential ongoing terrorist activities, track evidence on the basis of attributes of the people and events being tracked. This year, ASAM was submitted to the RDEC Prototype Platform.

2. **Develop and/or adapt a dynamic social network modeling tool.** The Center for Analysis of Social and Organizational Systems (CASOS) at Carnegie Mellon University, under the leadership of Prof. Kathleen Carley, has developed a suite of organizational analysis tools that combine social network analysis with agent-based modeling capabilities to represent, measure, and predict interconnected networks of people, knowledge, resources, tasks, and organizations. One such tool is Organizational Risk Analysis (ORA). In the second year, ORA was evaluated on the RDEC Prototype Platform, and the feedback from that evaluation was incorporated into new releases of ORA. Plans are underway to evaluate ORA in RDEC's Data Protected Platform context.

3. **Provide an environment for integrating network modeling tools.** Two major requirements of the NEMESIS integration scheme are that it be able easily to accommodate the integration of additional network organizational analysis tools and that the integration scheme ensure that the tools analyze the same organization. We address the first of these with an architectural solution: bus-oriented integration. This approach leverages the small amount of effort needed to integrate a new tool into NEMESIS to provide immediate integration with the tools that have already been integrated. This approach also means that NEMESIS will need a common language to describe the network organizations. The second requirement above, though, means that this language cannot simply be a hodge-podge of different representational techniques, one for each tool. If this were the case, all tools could use the language but there would be no guarantee that they were analyzing the same organization. Instead, the *same* representation of a given organization needs to be usable by all integrated tools. In the second year, a new version of ODL was released that provided greater flexibility in expressing informal networks as well as formal organizational networks, and a Technical Integration Experiment (TIE) was performed with Cycorp that translated knowledge from their Terrorism Knowledge Base to ODL, which enabled the NEMESIS environment to transform it so that it could be used by ORA.

4. **Provide capability for collaboration in this environment**. NEMESIS presents an unusual collaboration context because collaborations will likely take place over an extended period of time. The work products of the collaboration, ODL descriptions of organizations, transactions, and documents associated with these organizations, will also evolve over time. In some cases, there will be multiple versions of those work products, some exploring speculative "what if" kinds of scenarios, others describing the team's evolving understanding of the situation in a sequence of revisions and versions. In addition, there is a strong need to understand the context of each document—who is the author, what sources were used, when was it first produced, who modified it last (and when, and why), what documents need to be revised given that there is a new version of another document, and so on. In the first year, the NEMESIS project specified and developed a capability called *collaborative versioning* to address these issues, and implemented a subset of it for prototype. This year, that capability was extended to include a dynamic and flexible version of collaborative workflow. The result, *collaborative versioning and workflow*, provides the benefits of collaborative versioning and facilitates multiauthor collaboration on documents in a highly unobtrusive manner.

5. **Provide a measurement framework for assessing the success of NEMESIS.** Measurement is an important aspect of NEMESIS. In the second year, we worked with representatives from SAIC to fit NEMESIS into the Topsail measurement model and began work on defining suitable metrics, measurements, and quantifiers. Aptima itself has a lot of experience measuring human performance, and we presented our approach to human performance measurement to SAIC personnel. We will be working to integrate both approaches for NEMESIS.

The second year of the NEMESIS project has extended the first year's solid foundation for developing a state-of-the-art environment that provides an integrated analytical and predictive view of hostile networks and in which collaboration among counterterrorism analysts is facilitated.

Collaborative versioning and workflow provides rich capabilities for managing a revisable collection of documents and will provide sophisticated but unobtrusive coordination of missions and tasks involving multiple analysts. NEMESIS' bus-oriented integration platform and accompanying SDK provides a foundation that will scale gracefully as new applications are added, and ODL provides a common language for integration, as illustrated by the TIE with Cycorp. The ASAM tool from the University of Connecticut provides an advanced means to identify matches between the pattern of transactions and a library of patterns of hostile activities, to estimate the likeliest current state of those patterns, to manage all the associated uncertainties, and to find optimal intervention strategies. The ORA tool from Carnegie Mellon University provides a means to understand the vulnerabilities of the hostile network organizations and, in conjunction with other tools from Carnegie Mellon, also to predict the effects interventions such as disabling or capturing given members in the network.

Subsequent iterations of NEMESIS will refine these capabilities with special attention to the specific requirements of the intelligence community and their collaboration and integration environments. To ensure that the product of this refinement is useful to the intended users, an informative and rigorous measurement program for NEMESIS will continue to be pursued in quasi-laboratory settings as well as quasi-field settings.

We believe the result will be a quantum leap in collaborative, automated capability for analysts. NEMESIS is poised to enable them to leverage each others' knowledge and skills more effectively with more effective center-edge collaboration, to work faster by automating the management of collaborative artifacts and what-if scenarios, and to work smarter by amplifying the analysts' cognitive processing of patterns of hostile activity and of hostile meta-networks, and of the benefits that can only be obtained from multiple, interactive views of the same situation

## Introduction

Terrorist organizations like al Qaeda operate as *networks* of shadowy, dynamic, globally distributed stateless individuals and groups, with few fixed assets or addresses, but with sufficient connectivity to achieve highly destructive coordinated actions. Recent advances in network analytic technology show promise for exploiting that connectivity to understand and monitor the hostile networks and to formulate and evaluate the effectiveness of strategies to disable or destroy them.

NEMESIS is aimed at leveraging those technologies to provide an integrated, extensible network modeling environment that can be used for collaboration among analysts. There are three main components to this environment:

- **A collaboration platform.** At one point, Groove Virtual Office from Groove Networks, Inc. had been selected as a collaboration tool; however, requirements of the program have changed, and NEMESIS is now beginning to understand what it will mean to integrate with Vignette Collaboration Server. Beyond that, however, there are special requirements imposed by the need to manage models, documents, and other artifacts of collaboration as analysts' understanding of a hostile network evolve.
- **An integration platform.** The new network analytic technologies display a surprisingly diverse set of approaches, assumptions, and world views, yet each can provide valuable tools that amplify human cognition and at the same time automate what can be automated. It stands to reason that an environment that integrates this diversity to provide multiple views of the same hostile network will also be valuable, especially if it is architected so that new network analytic tools are easily added as they become available.
- **The network analytic tools themselves**. NEMESIS initially focuses on two tools. One uses a transactional network modeling approach to monitor filtered transactions to detect potential matches against known hostile patterns; the other uses a dynamic social network analysis approach that analyzes organizations and their context to detect risks and vulnerabilities.

In addition to these components, it is important to be able to assess the effectiveness of the environment on an ongoing basis, both to provide a means to evaluate it and to provide a means to improve it. For this reason, NEMESIS will involve an ongoing measurement effort.

This report documents the results of the first two years' research, with emphasis on the second year. It describes the collaboration capabilities, the integration platform, and two state-of-the art network analytic tools as well as a prototype implementation and demonstration. This research continues the path of what we believe is will be a fruitful multi-year effort to make important and useful advances in the capabilities available to support analysts in their all-important missions.

## Technical Objectives

There are five major tasks involved in each iteration of NEMESIS:

1. Develop and adapt a functional transactional network modeling tool;
2. Develop and/or adapt a dynamic social network modeling tool;
3. Provide an environment for integrating network modeling tools;

4. Provide capability for collaboration in this environment; and
5. Provide a measurement framework for assessing the success of NEMESIS.

Aptima is primarily responsible for tasks 3, 4, and 5, and our partners are responsible for the other tasks. Staff in the Electrical Engineering and Computer Science Department at the University of Connecticut are primarily responsible for task 1, and staff at the group for Computational Analysis of Social and Organizational Systems (CASOS) at Carnegie Mellon are primarily responsible for task 2

We now discuss second-year activities and accomplishments for each of these tasks.

## *Task 1. A Functional Transactional Network Modeling Tool: Adaptive Safety and Monitoring (ASAM)*

### Introduction

The ASAM system is an advanced counter-terrorism analysis tool designed to have the following capabilities:

1. Predicting intent and future states of the terrorist activities: The ASAM system employs a novel combination of hidden Markov models (HMMs) and Bayesian networks (BNs) to compute the likelihood that a certain terrorist activity exists. This likelihood is an important indicator of terrorist threat.
2. Identify threats: The ASAM system utilizes attribute-aided tracking and hidden Markov models to identify suspicious activity consistent with an *a priori* terrorist template model. A probabilistic matching of modeled attributes with the observed attributes provides an ability to identify the suspicious person, place, or object.
3. Options analysis: The ASAM system can suggest actions to prevent the terrorist activities. Using optimization techniques, effective action sequences can be suggested. Therefore, the ASAM system increases the range of options and early alarms to facilitate preemption.
4. "Inverting the bath tub" and automation: The ASAM system provides efficient and effective methods for counter-terrorism analysis. "Inverting the bath tub" refers to upending the plot of the time an intelligence analyst spends on the functions of collecting, analyzing, and reporting information. That is, currently intelligence analysts spend the majority of their time on collecting and reporting information, when it should ideally be spent on analysis. The ASAM system is a semi-automated system, which has the ability to detect and track terrorist activity and perform what-if analyses to enable an analyst gain deeper insights into a potential terrorist activity.
5. Model and scenario generation: The ASAM system provides a means to develop models based on real world events. We have developed an Indian Airlines hijacking model and an Athens 2004 Olympics threat model. Using the ASAM system, potential threat scenarios can be built and used to suggest priorities for efforts to reduce the overall threats.

The ASAM system has a hierarchical process, where the lower levels correspond to HMMs, and the higher levels are modeled via BNs. These, in turn, can be hierarchical as well. Briefly, a HMM is a stochastic model used to evaluate the probability of a sequence of events, determine the most likely state transition path, and estimate parameters which produce the best representation of the most likely path.

Mathematically, a discrete HMM is described by three parameters: $\lambda = \{A, B, \pi\}$. Here, $A$ represents the transition matrix of the underlying Markov chain, $B$ denotes the probability of emission of a certain "symbol" from a particular state, and $\pi$ represents the initial probability distribution of the underlying Markov states. The BN is a directed acyclic graph (DAG) that consists of nodes and links. It formally represents an intuitive and modular representation of knowledge through causal links among nodes.

In this paper, it is assumed that the observed data (a series of transactions) is available from an intelligence database; it represents any kind of travel, task, trust, or communication between any person, place, or item of suspicious origin. As more transactions are detected, more links representing the transactions are made in the transaction space. The idea behind using an HMM is that we can represent its underlying states as snapshots of the growing transaction space, and that it is the *evolution* of these snapshots that provides the most valuable clue. Note that, within each of the states of the HMM, is a graphical representation of the terrorist network's activity. HMMs function in the transaction space under a fast time-scale, while BNs operate in the strategy space under relatively slow time-scales. Each HMM can be viewed as a detailed stochastic time-evolution of a particular node state represented in BNs. The HMMs send soft evidence to BN nodes, and the BN inference algorithms integrate the soft evidence from multiple HMMs into an overall assessment of terrorist threat. In other words, the BN represents the overarching terrorist plot and the HMMs, which are related to BN node, represent detailed terrorist subplots.

## Modeling of a Terrorist Event

In order to detect terrorist activities, the ASAM system must be given *a priori* information about the potential terrorist activities ("template models"), which are to be monitored. This *a priori* information is provided in the form of HMM and BN models of the terrorist activities. Examples of these models are discussed in later subsections.

Predicting a terrorist event out of vast amount of information is analogous to finding a needle in a haystack. While developing a model of a specific terrorist event out of the available information, one key question is: how much *a priori* information is needed to develop a good model? In analogy to the needle in the haystack problem, the question can be asked how big the magnifying lens should be in order to find the needle. The correct amount of *a priori* information in the model ensures a good design of a magnifying lens. Another issue which arises is the estimation of model parameters. In this case, a relevant question to ask is: How do we specify HMM and BN parameters? An approach to obtain the HMM parameters could be estimating it using Baum-Welch algorithm and maximum-likelihood estimation using historical data. When the historical data is not available, the parameters can be specified according to the model and the state description. For example, if the number of transactions in a state is high, then it is highly probable that HMM stays in that state for a long time. Similarly, if relatively few transactions are related to the state, then the probability of remaining in that state is low. Transition probabilities do affect the detection scheme; hence, the probabilities which best fit the scenario should be specified.

The ASAM system requires that the model be generic so that it can be easily instantiated for any specific name, place, or item related to terrorist activities. In the next section, we present an example to analyze the vulnerabilities of the Athens 2004 Olympics.

*BN Model of Terrorist Attack*

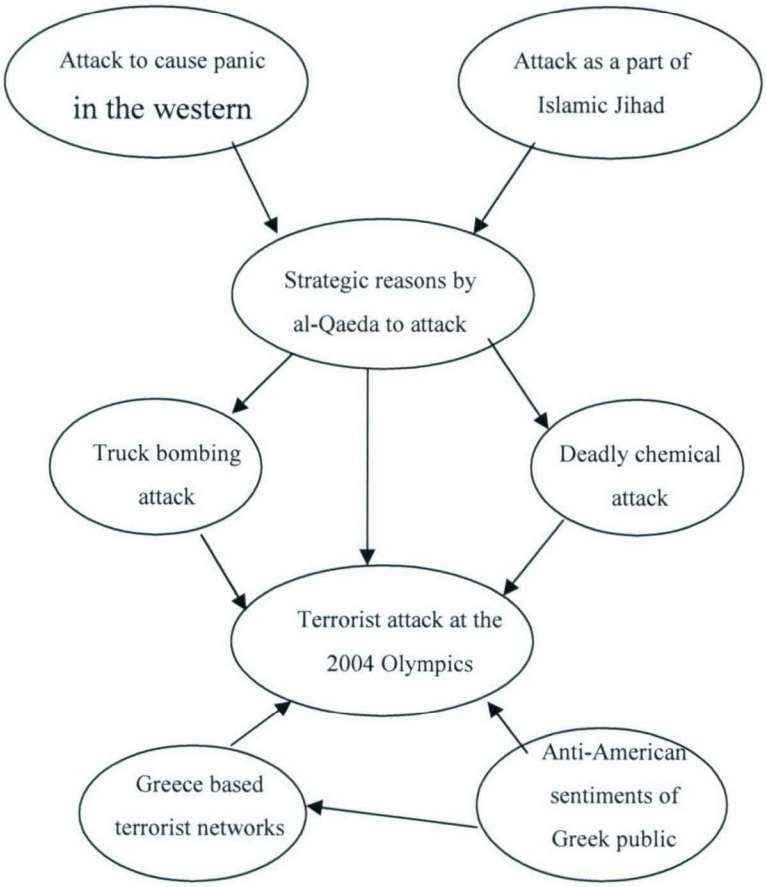



Figure 1. Abridged version of Greece Olympics model

The Athens Olympics is one of the biggest events of 2004. In this event, nearly 16000 athletes from 202 countries participated along with millions of visitors, volunteers, state officials, and dignitaries. Given such a mega event, any terrorist attack would be able to capture media attention across the world.

The BN model of vulnerabilities at the Athens 2004 Olympics is a collection of diverse potential terrorist targets and scenarios. One of the keys to the scenario is the geographical location of Greece and its proximity to Middle East and Europe, which could be an advantage for terrorist groups to penetrate and execute an attack. Greece is also prone to attacks from home-grown terrorist groups. The ongoing conflicts in Iraq, Israel, and Palestine also generate a vulnerable environment that could cause a significant threat to athletes and visitors from the USA, UK, Israel, and their allies. The construction delay in the Olympics sports complex was another problem that could leave many loopholes for terrorists to execute an attack. The BN model assimilates all the above-discussed scenarios and threats.

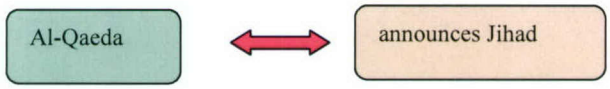
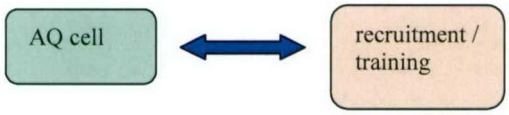
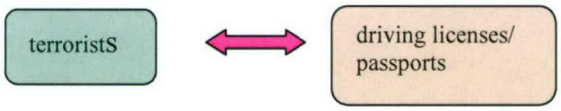
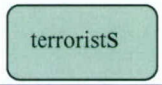
Figure 1 shows an abridged version of the BN of the terrorist attack threat in the Athens 2004 Olympics. The Bayesian node '*Strategic reasons by al-Qaeda (AQ) to attack'* depicts reasons such as getting attention throughout the world during the Olympics and causing panic in the western world. Another BN node depicts the threat of terrorist attack due to home grown terrorist networks

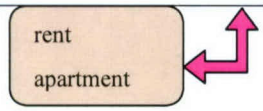
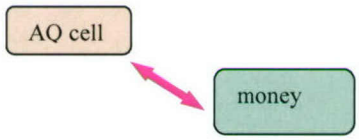
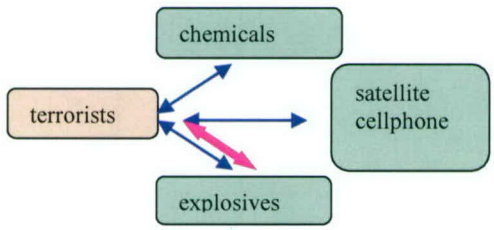
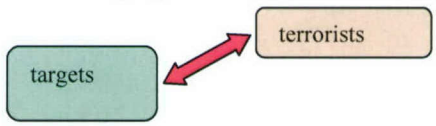
in Greece. The states of BN nodes '*Truck bombing attack*' and '*Deadly chemical cloud attack*' are modeled by the underlying truck bombing HMM and deadly chemical cloud HMM, respectively.

*Truck Bombing (HMM 1)*

This model presents a fictitious story that AQ and its affiliated terrorist groups were planning a truck bombing in Athens during the 2004 Olympics. Figure 2 shows the Markov chain of HMM 1. The HMM 1 consists of 9 states; the transition probabilities are shown next to the transition. A detailed description of the HMM states and transactions is presented in Table 1. The bulleted items in Table 1 show the transactions that characterize each state.

Table 1. Transactions for the Truck Bombing HMM

| State | Transactions |
|---|---|
| 1 | *AQ announces attack on western targets*:<br>• Al-Jazeera, a Middle-East based media, reports that AQ website announces an attack on western targets.<br>  |
| 2 | *Recruitment/ training of new members*:<br>• The ring leader in AQ recruits terrorists to carry out the truck bombing attack<br>• AQ cell recruits operators to execute the attack and drive the vehicle.<br>  |
| 3 | *Arrange driving licenses and passports:*<br>• The terrorists are embedded in Greece a few months or a year before the Olymics and set up the cell. AQ ring leader assigns the operators, planners, and facilitators for the attack. The facilitator provides driving licenses, passports, etc. to the operators.<br>  |
| 4 | • AQ cell members rent two or three apartments and they pay rent by cash.<br>  |

| | |
|---|---|
| |  |
| 5 | *Money for operation*:<br>• The AQ ring leader sends money to the AQ cell members via messengers.<br>  |
| 6 | *Gather Resources*:<br>• Terrorists purchase or steal chemicals, blasting caps, and fuses for explosives in Turkey and transfer via trucks to Greece. Terrorists purchase or steal respirators and chemical mixing devices and electronic parts such as satellite cellular phones from the illegal sources.<br>  |
| 7 | *Target reconnaissance*:<br>• Suspicious persons (bomb building experts, persons in the watch lists) reconnaissance the potential targets. Terrorists perform dry runs of routes to identify speed traps, road hazards, etc.<br>  |
| 8 | *Weapons installed*:<br>• Terrorists rent a truck. Terrorists modify the truck to handle heavy loads and neutralize any security arrangements at the target.<br>  |
| 9 | *Attack*<br>• The terrorists drive the truck into the target and detonate the bomb. |

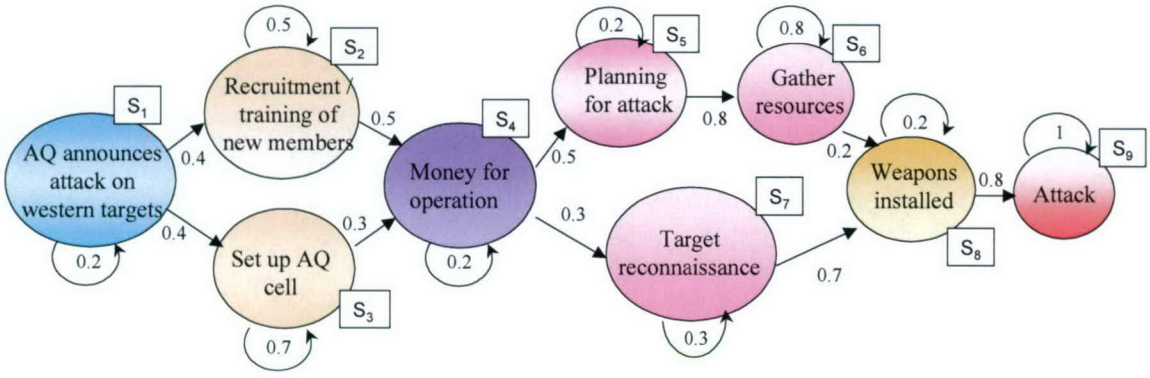



Figure 2. Markov Chain of Truck Bombing Model

*Deadly Chemical Cloud (HMM 2)*

This example depicts a hypothetical deadly chemical cloud attack. AQ plans a chemical cloud attack in a closed place through a subway ventilation system or in an open crowded place, such as downtown. The attack involves mixing lethal chemicals including blistering agents to cause third-degree burns, nerve gas, and choking agents.

The Markov chain depicting HMM 2 is similar to HMM 1 shown in Figure 2. However, the transactions defining the states of HMM 2 are different from those discussed for HMM 1. This is a consequence of the fact that terrorists employ different tactics in order to carry out different kinds of attacks.

While analyzing the vulnerabilities at the Athens 2004 Olympics, we hypothesized that terrorists might plan and execute multiple attacks at the same time. In order to detect these multiple attacks, we needed to adopt advanced target tracking methods. In the next section, we discuss such methods and illustrate their functionality based on the two examples discussed in this section.

## Advanced Methods for Tracking Terrorist Activities

One of the key capabilities of the ASAM system is its ability to continually track many instantiations of terrorist activity in a cluttered environment. While the detection and tracking of a single terrorist activity using an HMM involves the forward or forward-backward algorithm, the competition amongst HMMs for the observations (i.e., the *association* of transaction observations to the HMMs whence they come) suggests that inference becomes essentially a multiple-target tracking (MTT) problem.

Traditional methods of tracking such as the multiple hypothesis tracking (MHT) and the Joint Probabilistic Data Association Filter (JPDAF) are not directly applicable to tracking terrorist activities due both to the models and to the nature of the observations. In this case, the observations appear to be superimposed: for example, the observations associated with HMM 1 overlap the observations associated with HMM 2. Superposition of observations related to both HMMs can be linear, as in power, or nonlinear, as in the case of an *OR* combination of the observations.

*Multiple Target Tracking*

As discussed above, the model for a particular terrorist network can be represented as an HMM. Suppose we want to detect the presence of either of the HMMs discussed above. The problem is complicated because it requires checking the existence of both HMMs. While we can assume that the HMMs describing these two terrorist activities are conditionally independent, we must however consider that their observation processes are strongly dependent. In order to compute the likelihood of multiple HMMs, we invoke a target tracking algorithm that assumes the HMM state sequences to be conditionally independent and their likelihoods to be conditionally dependent.

After evaluating the likelihood of each HMM (or combinations of HMMs) given the observations, we can then determine the validity of a hypothesis using a sequential probability ratio tests (SPRT) to update its track score. In this Page-like test, the track score of each hypothesis is compared to a threshold and if it rises above a threshold, then the hypothesis is confirmed and a new hypothesis is formed. The following section highlights the logic behind hypothesis maintenance and formation.

*Multiple Hypothesis Tracking*

When there is data association uncertainty (i.e., the observations are not labeled, and it is not known from which source, if any, a given transaction emanates), correct statistical inference requires the evaluation of all possibilities. An MHT (in the kinematic target context) is a type of target tracking system that forms alternative data association hypotheses every time an observation-to-track conflict arises. A special case of this is known as Reid's algorithm. After a new observation is made, a new set of hypotheses is created and is then propagated to the next scan. It is important to properly form and maintain track hypotheses, since their number can increase exponentially with each additional observation. In this section, we present an algorithm similar to Reid's, but from a track-oriented approach, and we adopt it from tracking targets to tracking transaction patterns.

For example, consider only two HMMs that describe the activities associated with HMM1 and HMM2. As shown in Figure 3, the MHT begins under the assumption that the two HMMs are independent. $H_0$ represents the null hypothesis, and $H_1$ and $H_2$ represent active hypotheses in a conventional detection problem. For example our first test, "Test #1", is trying to determine if HMM1 or HMM2 is active. If HMM1 is active, then the next test will be "Test #2" where the NULL hypothesis becomes the existence of HMM1 and the new active hypotheses are: 1) HMM1 and HMM2 are both active; and 2) nothing is active. If our detection algorithm receives a few transactions that strongly imply that HMM1 is currently active, then HMM1 will be confirmed (statistically) and our new hypothesis will become HMM1 and HMM2 are active versus only HMM1 is active. There are of course many different transitions between tests and these are represented by the arrows in Figure 3.

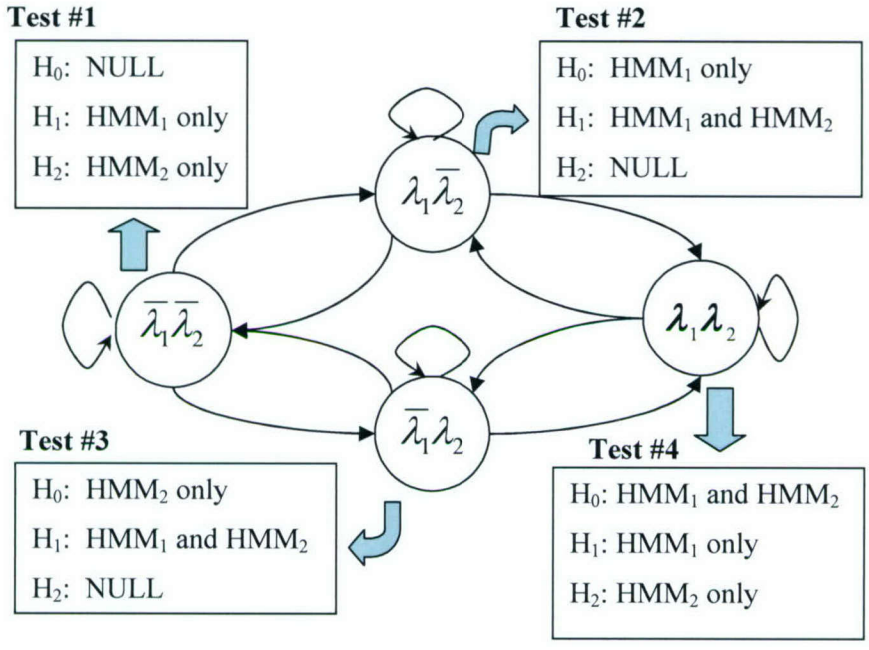



Figure 3. MHT for two HMMs

One of the benefits to this implementation of an MHT is that it is not susceptible to exponential complexity. This is because the number of hypotheses is limited by the number of HMMs which need to be tracked, and hypothesis generation is based on a logical combination of previous knowledge.

*Attribute-Aided Tracking*

Attribute-aided tracking is the process of collecting data about the features of a target from one or more sources to enhance the knowledge about the dynamics and class of the target. HMMs describe the dynamics of a terrorist network by including *a priori* information that describes the people involved, the temporal characteristic of the transaction, the event place, and other characteristics. These attributes are directly embedded within the underlying states of the HMM, and can be used to distinguish the targets of interest from ambient background noise.

For example, suppose that we are tracking HMM 1 as described in Table 1. The fourth state of HMM 1 contains a transaction related to the terrorist's arrival in Greece. Transaction and features related to this state are shown in Figure 4. In order to distinguish the terrorists from millions of visitors arriving in Greece, we must consider their attributes. If these men turn out to be around the ages of 50-60 and are citizens of a friendly country, then they are certainly less likely to be a threat than men around the ages of 30-40 yrs from nations tied to terrorists. It is for this reason that attribute fusion plays a pivotal role in the ASAM system. The main purpose behind attribute-aided tracking is that we can use it to refine our knowledge about a group or groups of terrorist cells.

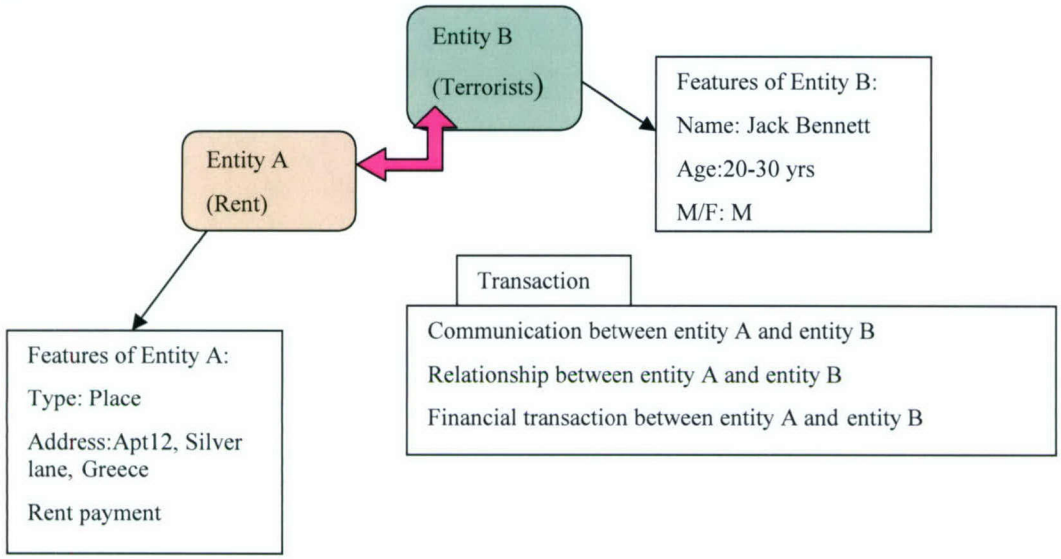



Figure 4. Transaction and Features

## Prototype Platform Participation

The ASAM system participated in the RDEC- Prototype Platform testing on December 9, 2004. This section describes various software modules which we developed or upgraded for PP testing.

*ASAM Control Center*

The ASAM control center consists of all the software related to the ASAM system. The user can execute it using the "ASAMCC.exe" file. The interface of the ASAM Control Center is shown in Figure 5.

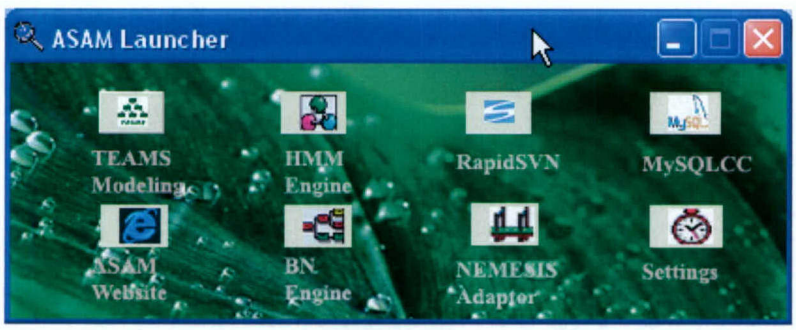



Figure 5. ASAM Launcher Interface

The first time the ASAM Launcher is opened the user should first click on "Settings" and specify the location of each of the installed components.

*Modeling in TEAMS*

TEAMS® (Testability Engineering and Maintenance System) is software developed by Qualtech Systems Inc. (http://www.teamqsi.com). The modeling interface of the ASAM system is currently developed on the TEAMS 7.0 beta platform.

The ASAM system uses TEAMs as a method for designing BN and HMM models. Other functions of the TEAMs software are not supported by ASAM. This version of ASAM allows one level for

the BN and one level for the HMMs. Each HMM is associated with one BN node (which has binary states: the HMM results are reported to one of the state and the other state is essentially the null hypothesis, e.g., the modeled terrorist activity doesn't exist or isn't activated). As an example, a BN model is shown in Figure 6. The model Transportation Attack is on the uppermost level and the three HMMs, "Collect_Resources", "Planning_And_Strategy" and "Preparations" (nodes with solid line box) are on lower level.

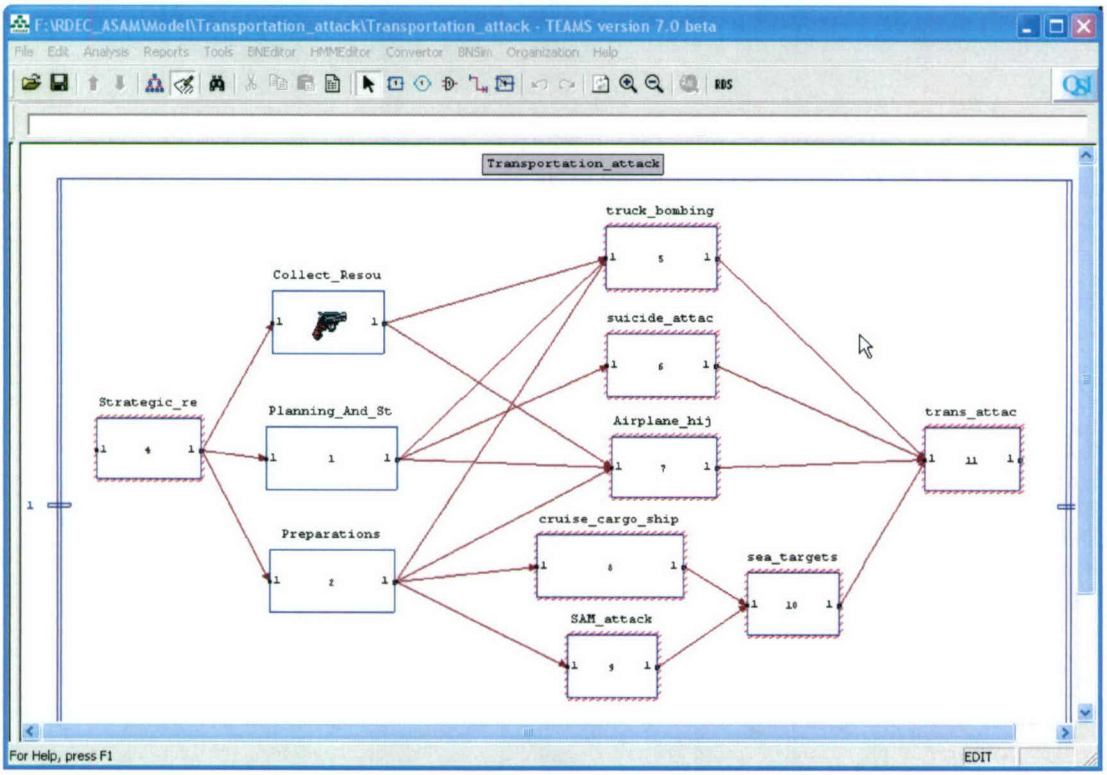



Figure 6. Bayesian Network Structure in TEAMS

This version of ASAM has limited compatibility with TEAMs so we will now discuss the functionality of ASAM within the TEAMs.

*Building a New Model*

**Step 1:** Developing a new model

By clicking the menu item: "File → New", the user will see a dialog. It is recommended that a new folder is created for each new model as there are a few files created for each model. Double click on the new folder, now a new model can be saved in this folder, e.g., "newModel". The default file type in TEAMS is *.tms.

**Step 2:** Designing a Model

The user should start by adding nodes in the newly created model. By clicking the "Add Module" on the toolbar, the user can see a dialog. Please type in the node name as desired and click "OK". Another dialog box pops up after specifying the node name. Just use the default values and click

"OK". A new module "node1" will then be added into the model. As shown in Figure 7, the user can see the model "newModel" has a module "node1".

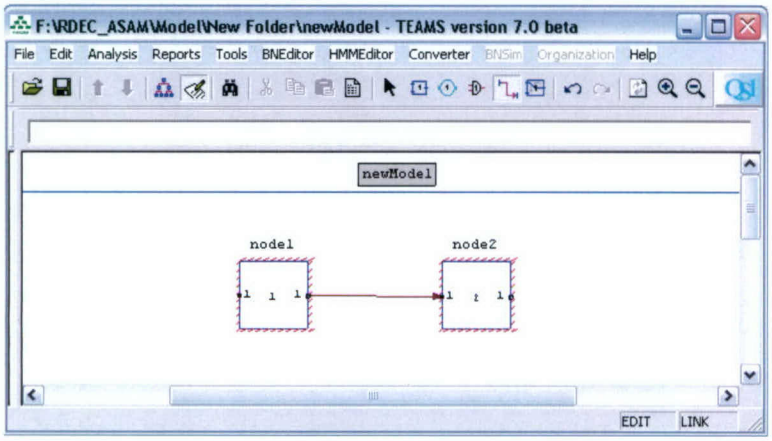



Figure 7. New Model Containing Two Modules

After the user has added more than one module, a link can be created between them by using the toolbar item "Add Link". In TEAMS, modules are structured from left to right. A link always starts from the right "port" of a module and ends at the left "port" of a module. It is important to note that ASAM interprets these nodes differently than the TEAMs software. In ASAM the nodes on the top level will represent BN nodes and nodes on the bottom level represent HMM nodes. If you click the "Select" button on a specific node and click the toolbar item "down a level", the user can view the bottom level and design the state sequence of a particular HMM. The links on the BN level show the causal relationships between the BN nodes and the links on the HMM level denote possible state transitions.

**Step 3**: Generate BN

After the structure of the ASAM model (both BN level and HMM level), the user should generate the BN conditional probability tables (CPT) ("BNEditor → generate → CPT"). By default, all the BN nodes are binary with states "Yes" and "No", and with equal probabilities. To change the causal relationship between nodes, first select a module with the pointer tool on the toolbar, and then select "BNEditor → Causal Relationship".

**Step 4**: Build a HMM

Before building a HMM, the user must first decide which BN node the HMM will report soft evidence to. It is important to note that HMMs act as "sensors" and BN nodes act as "decision makers" in that the HMMs actually "measure" and report terrorist activity while BNs are used to evaluate the probability of an attack by considering evidence from all BN nodes. To set one BN to receive evidence from a specific HMM, select "BNEditor → Node State" form the menu and edit one of the states associated with the HMM by editing the "underlying HMM model".

By doing this it is assumed that the user has already designed a specific HMM for this BN node. This can be verified by double-clicking on the BN node and check to make sure there is an HMM present on the "bottom level". Otherwise, the user can construct the HMM Markov chain on the lower level now. If the user decides to now build the HMM a message will pop up to remind the user to update the HMM related data such as prior probabilities and transition probabilities for the HMM states.

*View an Existing Model*

**Step 1:** By clicking on the menu item: "File → Open", and choosing the right "*.tms" file, the user will be able to load an existing ASAM model. Figure 4.1 is an ASAM sample model "Transportation_attack" loaded into TEAMS, where the three BN nodes with blue blocks have been set to receive evidence from HMM models.

**Step 2:** Click the menu item: "Converter → Load ASAM Model", now the user will be able to access the ASAM specified model.

**Step 3:** Use the menu list "BNEditor" to see the BN related model information. Be cautious with the first item "Generate CPT", the user will see an alarm message telling the user that the model already exists, if the user regenerate it, the user will lose the previous inputs, i.e., all the conditional probabilities will be set back as default values. Click the menu item "Node State" (if this item is inactive, please select a BN node first) and the user will see a dialog as shown in Figure 8. This dialog allows the user to edit the states of the BN nodes as well as the BN-HMM relationship. The example of Figure 8 is for the BN node named "Collect_Resources" which has binary states "Yes" and "No". The state "Yes" is set to receive evidence from the HMM "collect resources". The "Description" in the table is for the corresponding HMM model and the "Description" with the edit box is for the detailed description of the BN node. We currently only support the "CPT" type of causal relationship. Currently, each BN node allows only one HMM to report evidence to it. But the user can always achieve multiple hypotheses by setting up more nodes in the BN level.

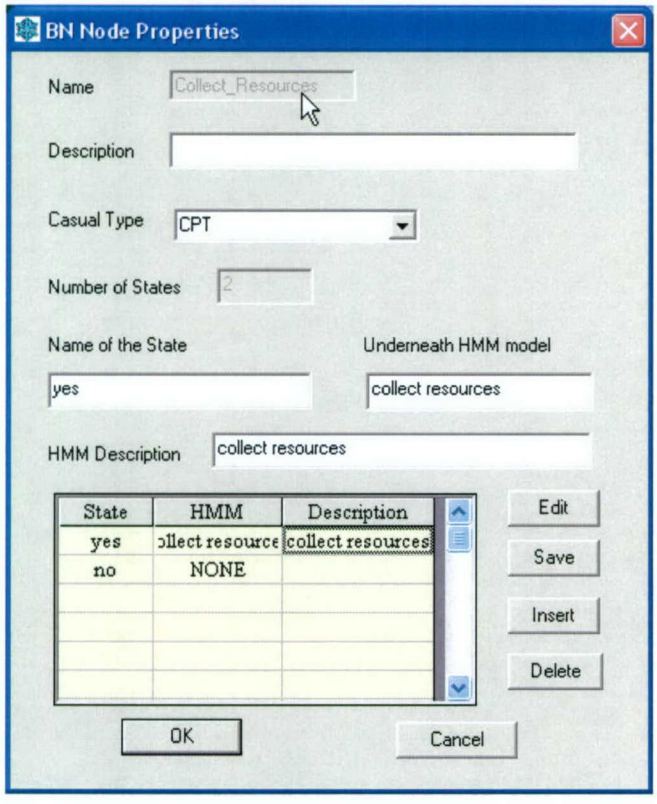



Figure 8. BN Node Properties

**Step 4:** Use the "HMMEditor" function from the menu to see information related to HMM model. Choose the BN node which has a HMM model underneath, and click "Down a level" (downwards arrow) from toolbar, the user will reach the HMM level which shows the Markov chain on the window (Figure 9 as an example). Although not drawn, it is assumed that each HMM state has the capability to transition back to itself.

Click on the menu item "HMM Model" and the user will see the definitions of the prior probabilities for each state associated with this HMM. It is possible to edit these probabilities by double-clicking on the grid, click "OK" when done. By default the probabilities are set to be uniformly distributed. The selected module will be highlighted on the screen when the user clicks on the corresponding state.

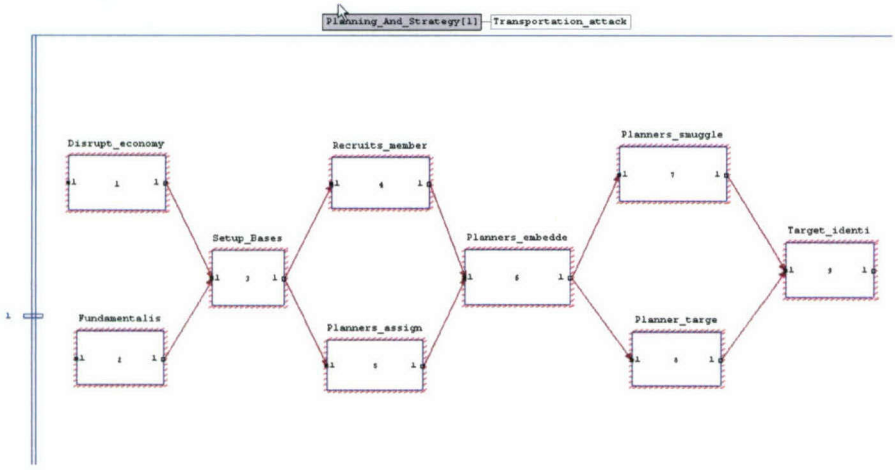



Figure 9. Markov Chain of the HMM Model

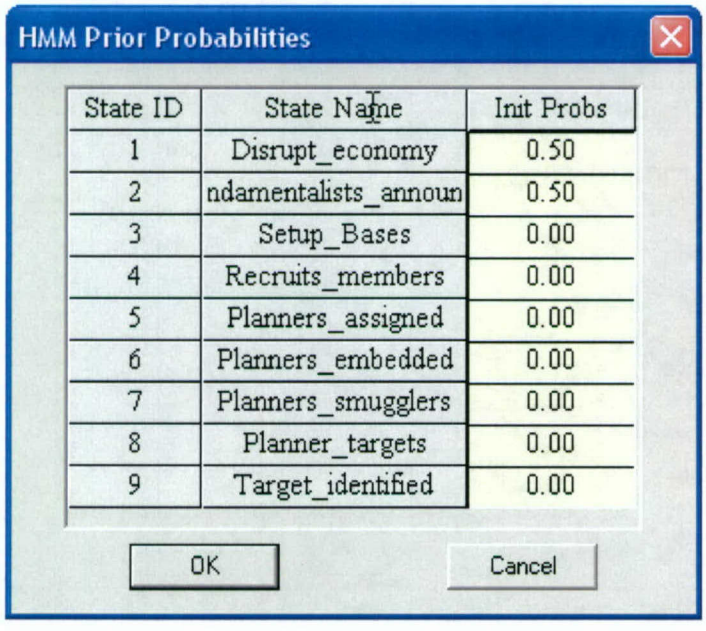



Figure 10. Prior Probabilities for Each State

Click the menu item "Markov Chain" to edit the transition probabilities between states. When the user focuses on an item on the table, a link or a module will be highlighted. The former case means that the transition is between different states while the latter case means the transition is back to the same state.

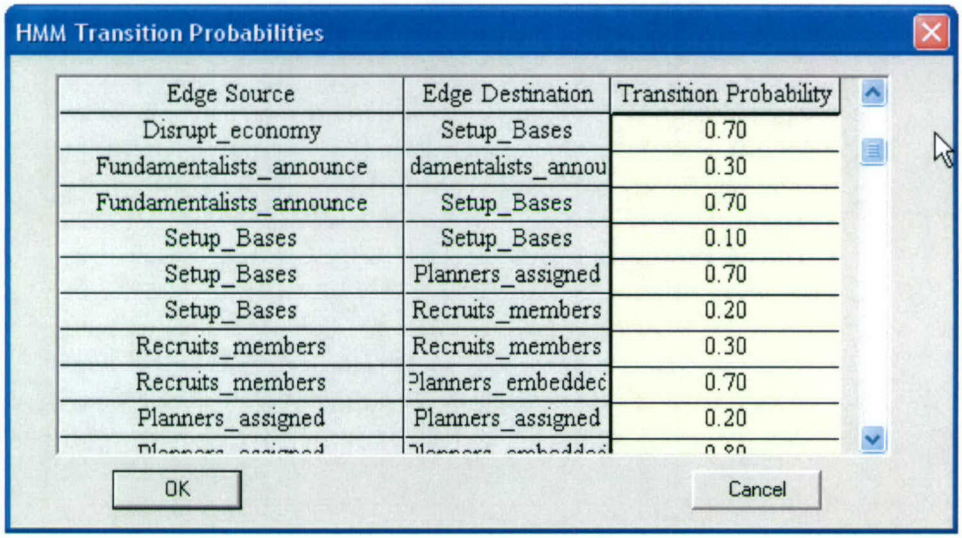



Figure 11. Transition Probabilities for the HMM States

Click the menu item "HMM Nodes" and the dialog box appears which allows the user to input all the nodes (attributes) associated with the HMM model. They are the transaction sources or destinations (transaction bindings). These nodes can be persons, places, or things. Different features associated with weights (means how important this feature item is) are available for each type of node. "Binding ID" can be a brief version of the node name.

After inputting all the nodes, it is possible to edit the transactions associated with each state of the HMM model by focusing on the state node and clicking "HMM Transactions" from the menu. A dialog, as shown in Figure 12, will pop up and the user can choose the source node, destination node, and transaction type from an available list, and insert the transactions into the list.

**Step 5:** Up to now, the user has gone through all steps of the ASAM modeling. If the user wants to analyze this model, save the model to database by clicking the menu "Converter→ Model to DB". All the necessary model information will be available in the database for further analysis. While doing that, the user will be informed about the database connection information and messages such as "This model already exists, does the user want to replace?" or "Model successfully saved in the database!".

**Step 6:** By default, the BN (*.dsl) and HMM (*.hmm) model will be hosted in the same directory as the TEAMS model. While saving to the database, all the HMM information and part of the BN information are saved into the database. In order for the BN inference engine to automatically switch between multiple models, please manually copy the *.dsl file with has the same name as the TEAMS model (e.g., "newModel.dsl") into a directory such as "F:\RDEC_ASAM\MODEL\". This directory will actually host all the *.dsl file for the available model

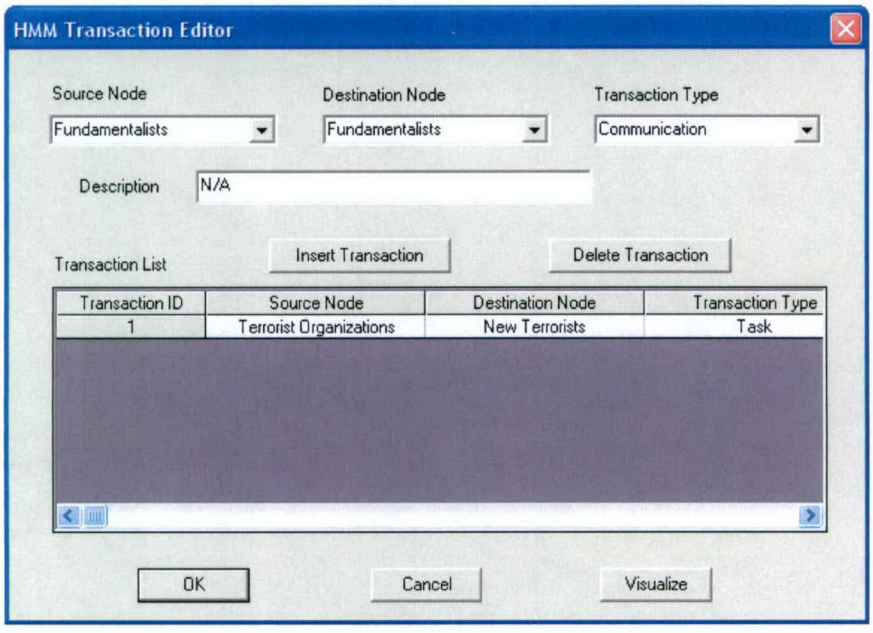



Figure 12. HMM Transaction Editor

## ASAM Website and Execution

The ASAM website provides a capability to view and analyze the results of the ASAM system. It is designed to visualize the results generated by HMM and BN software. Figure 13 shows the login page of the ASAM website. If login is successful, the website shows a page with brief description about the ASAM software . Next step is the selection of a model and mode of operation . The ASAM website provides two types of analyses:

- *Real time analysis*: It considers that transactions are available in real time.
- *What-if analysis*: It allows editing or adding the transactions

It is recommended to analyze real time results before performing what-if analysis. After selecting the real time analysis mode, the next page explains the types of results which the ASAM software can provide to the user. If user selects "local evidence" option then it brings a webpage which shows the available HMMs in the ASAM repository. Local evidence is the likelihood of existence of a specific terrorist activity. Figure 14 shows the local evidence associated with various HMMs. The website shows the existing results. New simulation can be carried out by clicking on "Initiate Simulation" button.

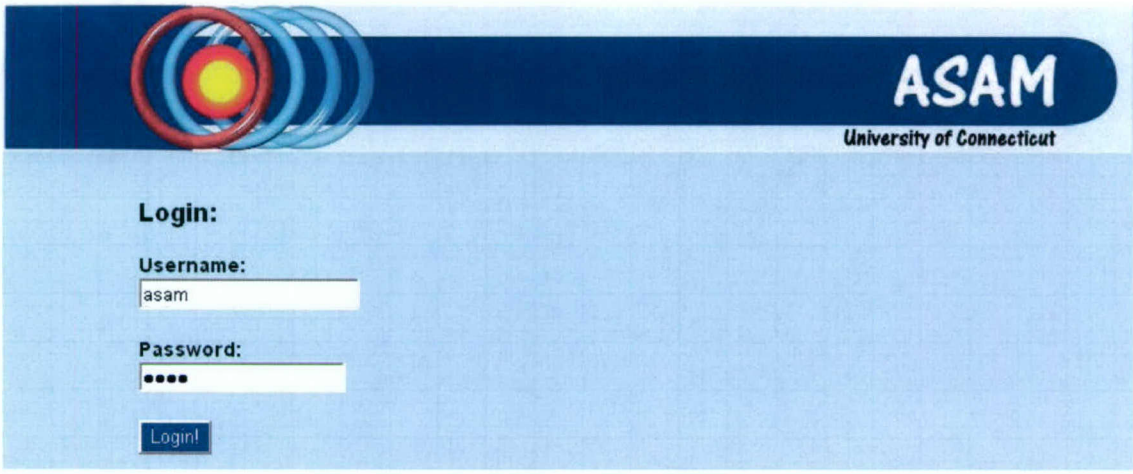



Figure 13. The ASAM website

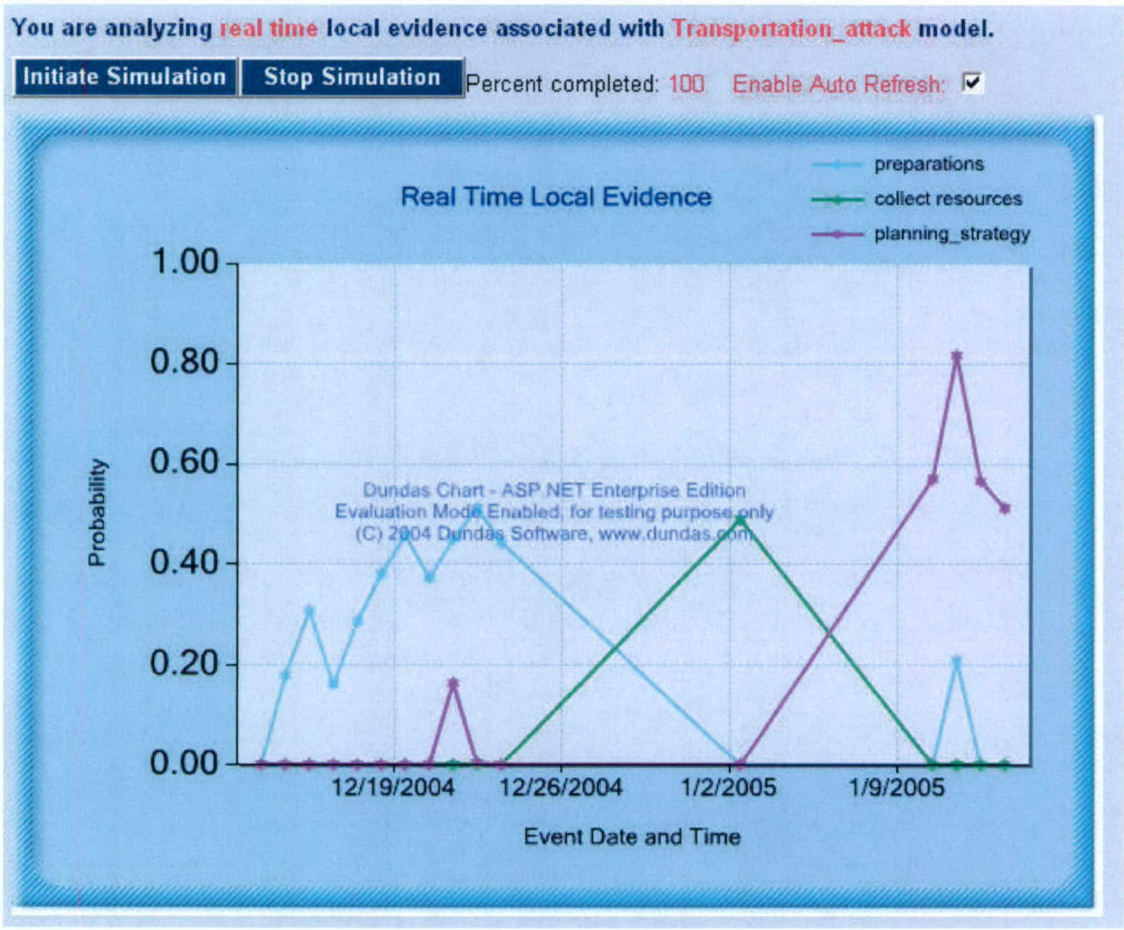



Figure 14. Real time local evidence

The ASAM website also reports the transaction details associated with suspicious transactions. These detected transactions can be viewed by clicking on "detected transactions." Figure 15 shows the detected transactions. The header of the detected transactions table provides capability to sort

the transactions alphabetically or chronologically. The transaction table has a transaction ID which is a hyperlink column. When the user clicks on the transaction ID; it shows the detailed features (attributes) associated with that transaction ID.

| Event time | People/person (source) | People/person (destination) | Transaction type | Event Description | Transaction ID |
|---|---|---|---|---|---|
| 12/7/2004 12:00:00 PM | Abdul Rahman Yasin | al Qaeda | Captured | NA | 273 |
| 12/7/2004 12:00:00 PM | Abdul Rahman Yasin | al Qaeda | Traveled to the US | NA | 274 |
| 12/12/2004 12:00:00 PM | Abdul Rahman Yasin | Al-Gamaa al-Islamiyya(Islamic Group, IG) | E-mail | NA | 288 |
| 12/14/2004 12:00:00 PM | Abdul Rahman Yasin | Al-Ummah | Phone Call | NA | 294 |

Figure 15. Detected transactions.

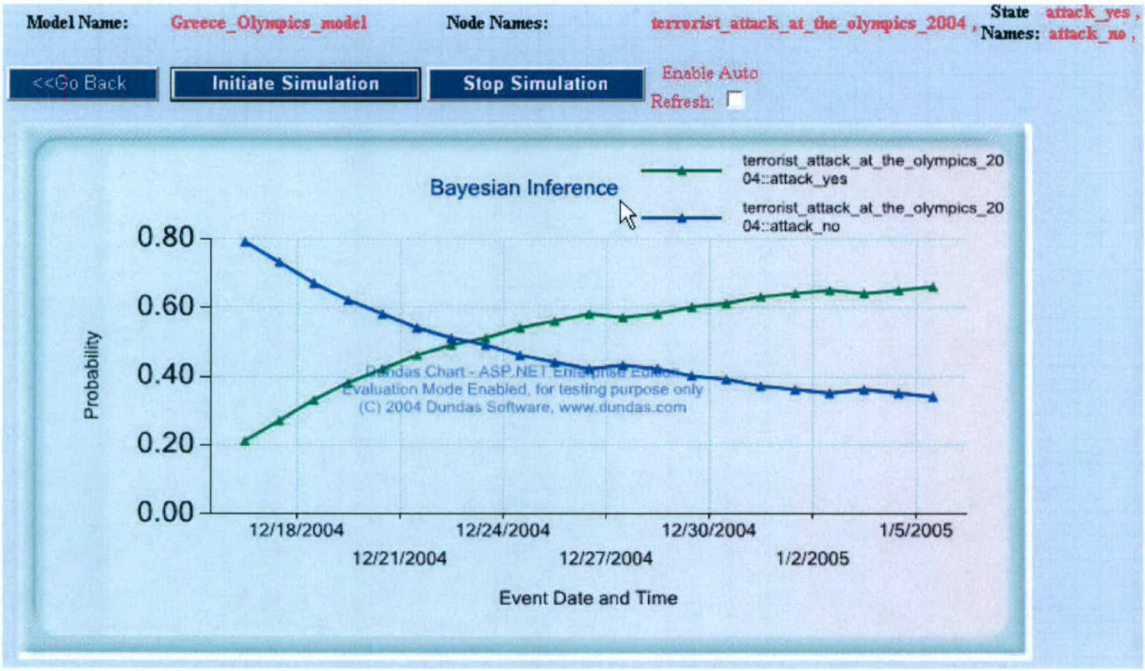


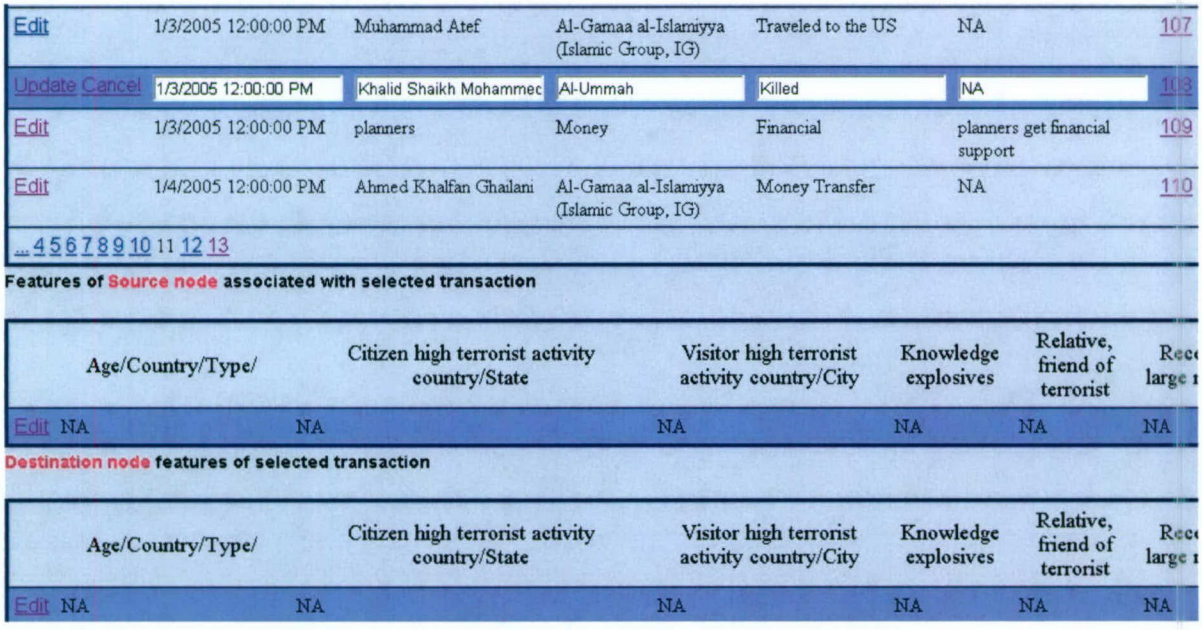

Figure 16. Bayesian inference

Next we analyze global inference associated with Bayesian network. Here user needs to specify nodes and states which he wants to analyze. The ASAM website shows the global inference for the

selected Bayesian nodes and states. Figure 16 shows the global inference for the transportation attack.

*What if Analysis*

The ASAM website provides capability to edit and update transactions. Figure 17 shows the process of editing a transaction. Features associated with transactions can also be edited.



| | | | | | | |
|---|---|---|---|---|---|---|
| Edit | 1/3/2005 12:00:00 PM | Muhammad Atef | Al-Gamaa al-Islamiyya (Islamic Group, IG) | Traveled to the US | NA | 107 |
| Update Cancel | 1/3/2005 12:00:00 PM | Khalid Shaikh Mohammec | Al-Ummah | Killed | NA | 108 |
| Edit | 1/3/2005 12:00:00 PM | planners | Money | Financial | planners get financial support | 109 |
| Edit | 1/4/2005 12:00:00 PM | Ahmed Khalfan Ghailani | Al-Gamaa al-Islamiyya (Islamic Group, IG) | Money Transfer | NA | 110 |

... 4 5 6 7 8 9 10 11 12 13

**Features of Source node associated with selected transaction**

| | Age/Country/Type/ | Citizen high terrorist activity country/State | Visitor high terrorist activity country/City | Knowledge explosives | Relative, friend of terrorist | Rec large |
|---|---|---|---|---|---|---|
| Edit | NA | NA | NA | NA | NA | NA |

**Destination node features of selected transaction**

| | Age/Country/Type/ | Citizen high terrorist activity country/State | Visitor high terrorist activity country/City | Knowledge explosives | Relative, friend of terrorist | Rec large |
|---|---|---|---|---|---|---|
| Edit | NA | NA | NA | NA | NA | NA |

Figure 17. Editing transactions and associated features

*Input transactions*

The ASAM website provides a user interface to insert new transactions. Each transaction can be categorized into three categories namely source, transaction type and destination. For example if a person buys chemical weapons from underworld mafia. Then person is source, underworld mafia is destination and transaction type is purchase and financial. Figure 18 illustrates the process of adding a new transaction.

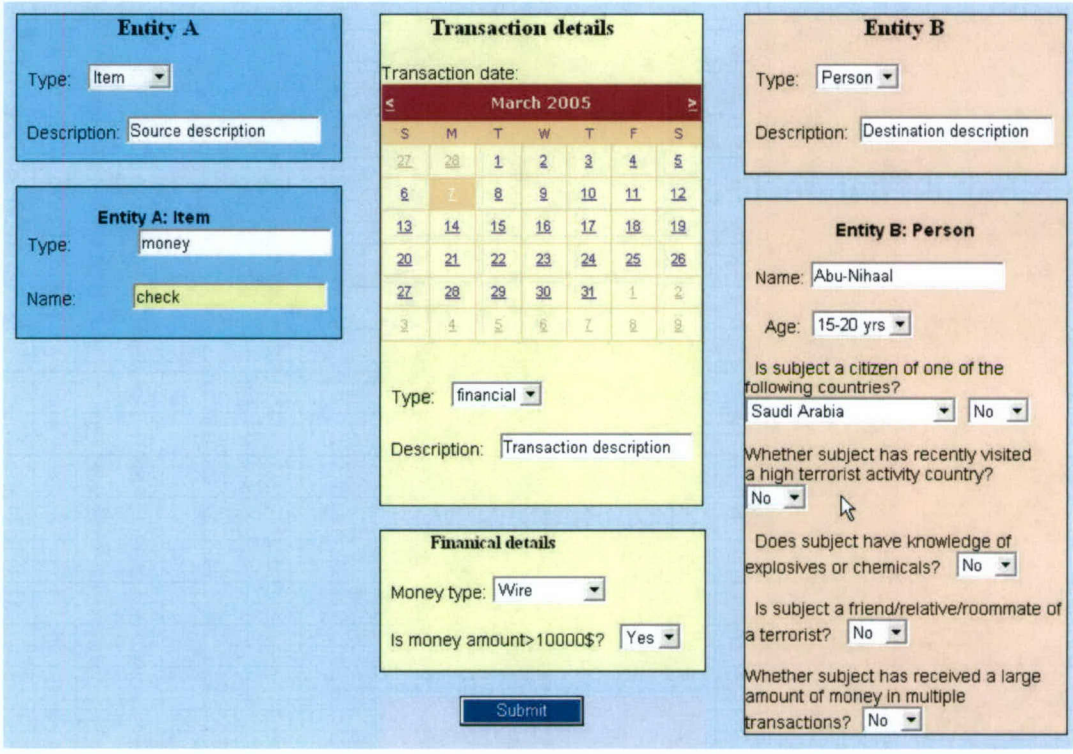



Figure 18. Input a new transaction

## Plans for Year 3

*Web Services Architecture*

Currently the ASAM system is implemented as a message-oriented architecture. The modules within the ASAM system communicate via the message bits stored in the database. The architecture is appropriate for a single user, but needs to be extended to multiple users. Hence the focus of the ASAM system is to adopt a web service architecture that can manage multiple concurrent ASAM sessions.

*Facilitate Inegration*

The web service architecture of the ASAM system will also facilitate integration with other tools. ASAM will respond to SOAP requests via its planned Web Service module, and ASAM in turn will be prepared to make SOAP requests of other applications, including the NEMESIS integration server.

*Preparations for DPP*

On the DPP, the effort will emphasize demonstrating ASAM against a more active data set, providing measurable results that demonstrate the capabilities. Hence there is a need of a good scenario. An elaboration of the Coronado terrorist attack model could be a suitable example for DPP.

*Visualization*

- Use Java technology to visualize the transactions as a graphical model

- Show some kind of game/animation along with graphs on the ASAM website to show that simulation is in progress
- Provide capability to compare what-if and real time results.
- Provide capability to enter probability of false alarm ($P_{fa}$), probability of miss detection ($P_{md}$) and other advanced options from the website.

*BN Software*

- Upgrade BN Engine to handle multiple users/sessions
- Update BN Engine to support Oracle 9i database
- Alter BN Engine to communicate with web services protocols
- Upgrade BN Engine with influence diagrams
- TEAMS modeling with global course of actions for what-if analysis

*HMM Software*

- Design method for extracting the identities of (people, places, and things). Implement identity estimation algorithm
- Distinguish between false alarm transactions and good transactions and compare likelihood of a particular transaction and compare with a threshold (code, update and communicate new information with database, visualization)
- Allow HMM program to handle multiple users/sessions and alter HMM program to communicate with Oracle and other web services protocols.

*Research Goals*

- Explosion of transactions is a big concern. Complexity level of algorithms needs to consider. The ASAM system needs to handle entity-link data which is of order O(105) entities and O(106) links.
- Related research on detection networks and influence diagrams.
- Research on efficient ways to optimize the possible action strategies .
- HMM detectability study needs to be completed.
- Implement partially observable Markov decision process (POMDP) to suggest actions at HMM level.

## *Task 2. A Dynamic Social Network Modeling Tool: Organizational Risk Analysis (ORA)*

### Introduction

The Center for Analysis of Social and Organizational Systems at Carnegie Mellon University, under the leadership of Prof. Kathleen Carley, has developed a suite of organizational analysis tools that combine social network analysis (E.g., Wasserman and Faust, 1994) with agent-based modeling

capabilities (Carley et. al., 2003) to represent, measure, and predict interconnected networks of people, knowledge, resources, tasks, and organizations.

Carley (Carley, 2003) describes these networks with what she calls a meta-matrix (see Table 2). The rows and columns name important kinds of nodes in the social network sense, and the labels in the cells describe the kind of network that results from connecting a node of the type in the column to a node of the type in the row. Since the matrix is symmetrical, cells below the diagonal have been left blank.

Table 2. Organizational meta-matrix.

| | **Individuals** | **Resources** | **Tasks** | **Organization** |
|---|---|---|---|---|
| **Individuals** | Social network | Capability | Assignment | Influence |
| **Resources** | | Resources | Needs | Core Capability |
| **Tasks** | | | Precedence | Commitments |
| **Organization** | | | | Organizational Network |

Given a network describing an organization in this sense, CASOS has developed a number of tools to display, analyze, model, and make predictions about it. Table 3 lists and describes some of them.

Table 3. Selected organizational analysis tools from CASOS.

| | |
|---|---|
| **Construct** | Multi-agent modeling of groups and organizations as complex systems that captures the variability in human and organizational factors. Dynamic relationships among simulated agents are grounded in structuration theory which is the notion of construction and reconstruction of the social system through human interaction based on rules and resources. Changes in the social system are defined and analyzed through social network analysis. |
| **OrgAhead** | Organizational learning model designed to test different forms of organizations under a common task representation. Each member of the organization receives information from a subordinate or from the environment, makes a decision based on the information and what he or she has learned so far, and provides superiors with an answer to the decision. |
| **NetWatch** | Simulation tools that allow experimental approaches to generation of plans of attack informed by knowledge of the structure of covert networks and information gathering approaches available to law enforcement organizations. Use of simulation is of particular importance due to scarcity of real-world data, the secrecy of the organizations that are subject of the study, and very limited ability to conduct empirical testing of hypotheses pertaining to disruption of terrorist networks. |
| **DyNet** | Reasoning support tool for reasoning under varying levels of uncertainty about dynamic networked and cellular organizations, their vulnerabilities, and their ability to reconstitute themselves. The analyst can see how the networked organization is likely to evolve if left alone, how its performance is affected by various information warfare and isolation strategies, and how robust those strategies are in the face of varying levels of information assurance. |
| **ORA** | Risk assessment tool for locating individuals or groups that are potential risks given social, knowledge and task network information. First you use information about people, knowledge, tasks, resources, and organizations to "connect the dots." Then, ORA examines this network and finds those dots, those people, who represent a risk to the overall system. Individuals are risks, e.g., if their removal from the network would debilitate it (the critical employee) or if they were to feed false information to others. they could create havoc (the rumor monger). |

Of particular interest to the NEMESIS project is the last tool, ORA. Its purpose is to assess the level of possible organizational risk and the factors that contribute to this risk. All measures are based on the meta-matrix and take into account the relations among personnel, knowledge, resources and tasks. These measures are based on work in social networks, operations research, organization theory, knowledge management, and task management. Full details are available in Carley and Reminga (2004).

Organizational risk as assessed by ORA is the risk run by having a set of vulnerabilities due to any aspect of an organization's structure that might cause problems. Given an organizational description with a meta-matrix network, measures are calculated to detect these vulnerabilities and to categorize risk. ORA examines several categories of organization risk such as individual, resource, knowledge, and performance. For each category, multiple metrics exist and have been implemented (though some metrics measure risk in multiple areas.)

For example, in assessing individuals, a measure called Critical Personnel Risk is assessed. This is aimed at answering questions such as:

- Would the removal of one employee/member from the organization greatly affect its ability to complete tasks?
- Do employees/members tend to have exclusive access to knowledge, resources, or tasks?
- Do employees/members have exclusive access to, or are they the only gatekeeper for, particular groups or individuals?

There are clear applications of this measure to the analysis of hostile groups.

### Year 2 Activities

In the second year CMU was tasked to do the following:

- Examine whether ORA could be used to support collaborative teams. We tested ORA on data collected in another study on collaborative teams (a study of a department at a university). Results indicated the need to do the following activities:

  a) generate new grouping algorithms,
  b) increase the speed of existing grouping algorithms,
  c) auto-generate the ego-net reports,
  d) generate a new "management" report indicating the "health of the group."

  CMU then leveraged the work funded by another project to support the development of a new measure of SSA and other metrics that we will then collect into a management report.

- Support the inclusion of ORA in experiments. CMU ran ORA in experiment one – the PP experiment. Results from that experiment suggested that there were needs for:

  a) increasing the speed of various algorithms,
  b) augmenting the interface to provide additional help,
  c) adding more context information,
  d) providing facilities to deal with node attributes,
  e) providing features for weighting and editing networks,
  f) enabling the system to be more robust in the face of larger datasets

g) adding features to handle issues of information assurance; and
h) providing appropriate data for the next round of experiments.

In response to this CMU has:

a) made substantial performance improvements to ORA
b) added additional help and improved the interface from a human user perspective,
c) added more context information,
d) began specing what will be needed by way of attribute level analysis to support the end user,
e) added features for editing and combining networks,
f) made the system more robust for larger datasets[1]
g) provided sample proprietary datasets for testing,
h) generated some very large artificial datasets for testing.

- CMU worked with SMEs from the intelligence community to identify needed changes to ORA. This has led to a refinement of the Intel Report and the creation of the Ego-Net (sphere of influence) report.
- CMU provided some training on ORA and expects to do more in June, 2005. Much of this training was conducted for the intelligence community and personnel associated with the experiments. CMU also briefed ORA to assorted agencies that are part of the various experiments being run and is currently planning to support putting ORA in a suitable round of experiments.

## *Task 3. An Integration Environment for Network Organizational Modeling Tools*

Two major requirements of the NEMESIS integration scheme are that it be able easily to accommodate the integration of additional network organizational analysis tools and that the integration scheme ensure that the tools analyze the same organization. We address the first of these by architectural means, which provides for a solution that leverages the small amount of effort needed to integrate a new tool into NEMESIS to provide immediate integration with the tools that have already been integrated. We discuss the architectural solution in this section.

This means that NEMESIS needs a common language to describe the network organizations. The second requirement above, though, means this language cannot simply be a hodge-podge of different representational techniques, one for each tool. If this were the case, all tools could use the language but there would be no guarantee that they were analyzing the same organization. Instead, the *same* representation of a given organization needs to be usable by all integrated tools. The NEMESIS solution is an XML-based language called Organizational Description Language (ODL), which we also discuss in this section.

---

[1] The work under NEMESIS by CMU will leverage another project that is funding the testing and augmentating ORA on networks of $10^5$ and $10^6$ nodes. Those changes should be incorporated in 6 months.

## Architecture

*Bus Oriented Integration*

NEMESIS selected a bus-oriented integration strategy rather than a point-to-point strategy. The reason is scalability. When integrating $n > 2$ applications it pays to organize the integration in a bus architecture because only $n$ connections to the bus are necessary instead of the connections that would otherwise be required.

$$(n-1)+(n-2)+\ldots+1=\frac{n(n-1)}{2}$$

Clearly it is preferable to have a bus-oriented solution that scales linearly with the number of applications instead of a point-to-point solution that scales quadratically with that number. Figure 19 shows the contrasting approaches diagrammatically.

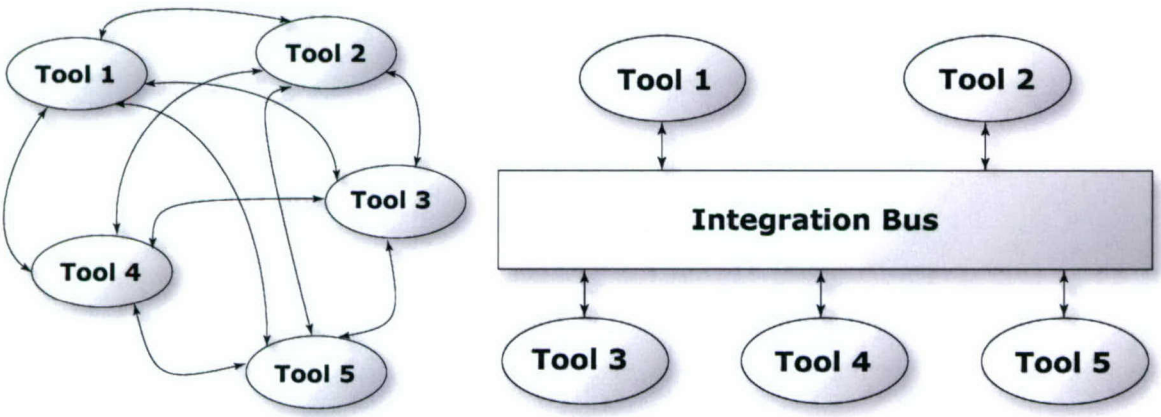



Figure 19. Integration approaches. (a) Pairwise integration; (b) Bus-oriented integration.

Still, there are difficulties and limitations associated with a bus-oriented integration. Precisely because point-to-point solutions are customized for every pair, deeper integrations are possible. Since communication between every pair will be unique anyway, developers can be very creative and thorough about allowing the pair to interact in every way they can think of. Bus-oriented solutions, on the other hand, need to identify ways of integrating that are common to all integrated applications. In many cases, their integration needs to be limited to the fact that they can operate on the same data. For NEMESIS, this translates into providing multiple views of the organization being analyzed. Multiple views provide considerable power, however, and the prospect of having to perform integration work on each of n-1 applications every time a new one is added will be daunting to organizations responsible for maintaining the system. The tradeoffs, therefore, lean heavily towards a bus-oriented approach.

There are three basic components to a bus-oriented integration solution:

1. A *common representation* of organizational networks that the applications associated with NEMESIS, such as modeling techniques, visualizations, and collaborative tools can use. Our choice was to develop and use an XML-based representation language called Organizational Description Language (ODL), described below.

2. The second necessary component is a *middle tier* with which to integrate. This is essentially a specialized Web server that "speaks" ODL. It is responsible for managing, storing, and manipulating ODL representations of the networks under analysis, and it will answer requests to retrieve or store a network description via a mechanism called *Web Services* or via simple file transfer.

   Web Services are really just a way of doing interprocess communication using the infrastructure of the Web. They are very tolerant of implementation differences, and applications integrated via Web Services need not be collocated on the same server, or even geographically. This opens interesting possibilities, for instance, if some of the network models will be run on supercomputers, or if there are licensing issues that prevent ownership but not use of an application.

   Simple Object Access Protocol (SOAP) is a common Web Services mechanism, but in this project, primarily due to our focus on collaborative versioning, Web Service functionality was implemented using the Web-based Distributed Authoring and Versioning (WebDAV) protocol.

   As work on Task 4 (collaboration) progressed, it became clear that the middle tier would have a dual purpose. It also is a server providing access to a collaborative versioning repository, as described below. Further, Web Services are a two-way street—that is, since both the server and the clients need to participate. Though it is straightforward to implement Web Services on the server side, the applications were either already developed without Web Services, as in ORA, or delayed implementing Web Services for reasons of focus and task prioritization, as in ASAM. The prototype demonstration therefore only used simple file transfer. But implementing Web Services requests to read and write ODL in the clients and the server will not be difficult in the future.

3. The final necessary component of a bus integration solution is a *software development kit (SDK).* The primary target of this kit is application developers, but for applications not under development, there is functionality for application users, as well.

   In general, applications need to read and write data from and to the NEMESIS repository. For applications under development for which Web Service requests are feasible, the SDK provides a windows control that can be incorporated into the application that makes it simple to do so. Even a Web Services client needs a certain amount of infrastructure to receive and request data, and if an application does not have that infrastructure, a different version of the Windows control can make a request of the repository to transfer a file via FTP from the server to a given location on the client.

   When Web Service or FTP requests are made, in some circumstances the data must be transformed before the application can use it. In general, the data in the NEMESIS repository is XML, so the SDK also includes an XSLT engine to invoke any required transformations (either as a part of the Web Service request or after the file has been transferred.) XSLT transformations can be sequenced in a configurable manner.

   The SDK also includes a small standalone NEMESIS launcher utility that enables these requests to and from the repository and that enables the transformations to be invoked. The launcher may also be configured to launch local applications of the users' choice.

Thus, the NEMESIS SDK provides a reasonably complete and flexible toolbox for applications and users to interact with the NEMESIS server.

### Repository

NEMESIS also provides a means to store and retrieve organizational models independent of any application. Without it, applications would need to be running simultaneously in order to be integrated, a major drawback. The repository provides for applications to be asynchronously integrated, that is, for them to provide different views of the same organizational from models stored in the repository rather than from models provided by other running applications.

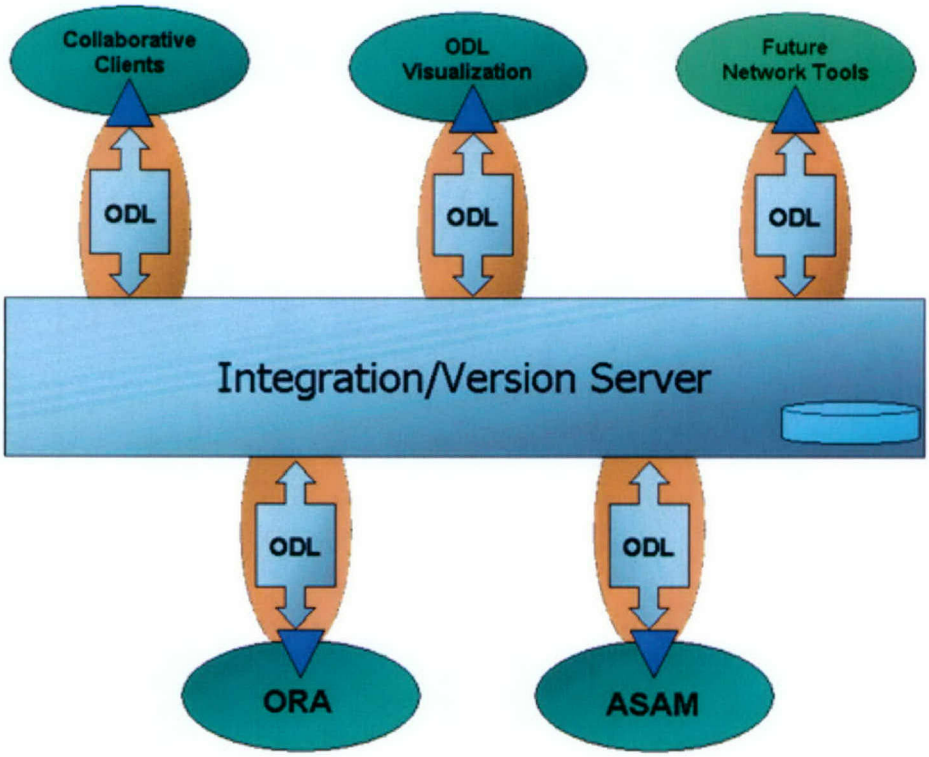



Figure 20. The NEMESIS integration environment

In NEMESIS as currently implemented, this capability is provided by a versioning repository that uses a database to store data. Repository functionality is currently integrated with integration functionality as well.

Figure 20 shows the NEMESIS integration infrastructure diagrammatically.

### Organizational Description Language (ODL)

In the second year, Aptima produced version 0.6 of ODL. The biggest change from earlier versions was that Networks were separated from Organizations so that informally defined networks could be expressed and so that Organizations whose network is not entirely understood could nevertheless be represented. Other changes involved adding IDs to Edges and limiting Deltas to organizations. What follows is a brief description of version 0.6.

ODL is a language for representing network-based descriptions of organizations intended to provide a *lingua franca* for integrating tools that provide network-based analysis of organizations. It is XML Schema based and has an intellectual heritage that includes two other XML based languages:

1. DyNetML from CASOS at Carnegie Mellon University (Tsvetovat, Reminga & Carley, 2003). ODL shares many characteristics with DyNetML. It is a platform for exploration of various aspects of organizational representation, however, so rather than impose quickly-changing and possibly evanescent requirements on DyNetML, a new language was created.
2. XHMTL from the W3C (W3C, 2004). XHTML is standard that describes an XML-based version of HTML. Unlike HTML, XHTML needs to work with other technologies that are also XML-based languages, such as Scalable Vector Graphics (SVG), Synchronized Multimedia Interaction Language (SMIL), X3D (and XML-based 3D file format), and many others. XHTML also has many other capabilities, not all of which are always needed. For these reasons, the W3C has focused on using XML language specification technologies such as Document Type Definition (DTD) and XML Schema to create modular languages. ODL has adapted this well-designed modularization technology to provide flexibility in the integration of organizational analysis tools.

We intend to adapt ODL to the needs of the broader organizational modeling communities in order to ensure that it is viable in the largest possible number of contexts.

Appendix B provides full, detailed reference material for ODL version 0.6. To help understand that material, this document provides a brief high-level overview of the language and the rationale for its features.

*The ODL Basics*

ODL describes organizations as networks of Edge elements that connect Node elements.[2]

Nodes are required to have a unique identifier, and have several optional attributes, such as Name, Node Probability, and the Date and time on which they begin and end applicability. Node elements may be of the several types

1. **Agent.** An individual person.
2. **Knowledge.** A text-based fact.
3. **Resource.** Money, supplies, weapons, etc. Used by instances of Agent and Organization, typically to perform a Task instance. A resource may optionally offer a Platform, which is a collection of references to other instances of Resource. An aircraft (with a collection of weapons) is an example of a Resource with a Platform.
4. **Task.** An activity performed by an instance Agent or Organization, often with help from one or more instances of Resource. A Task may have one or more Dependencies. Instances of Task may be organized higher-level Task by using Edges to express Supertask/subtask relationships.

---

[2] ODL is not intended to be an ontology in that it assumes the existence of, but does not define, a set of types. Many such types are subtypes of the Node or Edge type. In the discussion below, the phrases "X elements," "instances of X," and, when no ambiguity results, "Xs," are used interchangeably to refer to specific instances of ODL types.

5. **Event.** A meeting, an explosion, the broadcast of a message, or other one-time or recurring circumstance.
6. **Communication.** An information exchange between two or more instances of Agent or Organization.
7. **Location.** A physical place.
8. **Organization**. A known entity consisting of zero or more instances of Agent and/or Organization. It may, but need not, be described by an instance of Network (described below).

Nodes are connected by Edges, which have Source and Dest attributes that are references to unique IDs of Nodes. They also have attributes describing transition probability (the probability of traversing the edge), edge probability (that is, the probability that the edge actually exists), name, color (the type of edge, such as trust, influence, friendship, etc.), date/time when the edge begins and ends applicability, and zero or more instances of Annotation, which consist of arbitrary text. Figure 21 shows and example of some basic ODL.

```
<Model>
   <Agents>
      <Agent ID="A5" Name="Akhtar"/>
      <Agent ID="A10" Name="Latif"/>
   </Agents>
   <KnowledgeItems>
      <Knowledge ID="KHHC" Name="Hand-to-Hand Combat Skill"/>
   </KnowledgeItems>
   <Resources>
      <Resource ID="RBC" Name="Box Cutter"/>
   </Resources>
   <Organizations>
      <Organization ID="O4" Name="Harkat-ul-Ansar Mumbai Cell #1" Network="NHC"/>
   </Organizations>
<Networks>
      <Network ID="NHC" Name="HUM Cell Network #1">
         <Nodelist>
            <NodeRef Node="A5"/>
            <NodeRef Node="A10"/>
         </Nodelist>
         <Edgelist>
            <Edge Source="A5" Dest="A10" ID="E1"/>
         </Edgelist>
      </Network>
   </Networks>
</Model>
```

Figure 21. Sample basic ODL.

*ODL Networks*

Another subtype of Node is Network. An instance of Network shows the actual connections among Nodes. It describes relationships among the simpler kinds of Node instances just discussed. An Organization may or may not be described by a Network instance, and a Network instance may or may not describe an Organization.

A Network consists of a Nodelist, which contains a reference to each Node in the Network, and an Edgelist, which contains each Edge in the Network. A Network need not be connected in any particular way, or at all. Individual unconnected Node instances are sometimes called singletons, and they are easily accommodated within the Network description.

Because some applications involve small variants on large instances of Network, it is sometimes convenient to describe a Network in terms of its differences from another Network. This description is called a Delta, and the description is always of changes relative to a Network referenced in the required attribute AppliesTo.

*ODL Bindings*

So far, the representations described have been concrete—an Agent represent a known person, a Knowledge element represents an actual fact, an Organization represents a recognized network of Agent and Organization instances. There is a requirement that ODL represent partially or probabilistically known information as well. ODL meets this requirement with Binding elements.

A Binding element provides a place-holder for more concrete Node elements; it serves as a variable of type Node. Zero, one, or several Node elements may be attached to a Binding via its Value sub elements Value elements simply refer to a unique Node id. A Binding may have no Value elements attached, in which case it is simply indicating that an unknown node is participating in the network; it may have one Value element attached with some probability, in which case the Binding is "bound" to a Node with the given probability; or it may have several Value elements attached, each with its own probability, in which case the Binding's value is one of its Value elements, each with its specified probability. The sum of the probabilities of the Values attached to a Value must be less than or equal to 1. A Binding with a single Value attached with probability 1 is fully equivalent to the Node element to which the Value element refers. A Binding element has a unique id, and may be used anywhere a Node element may be used.

The semantics of Bindings can be clarified with a few examples.

- A phone conversation for which only one participant is known can be represented by a Communication element with the From attribute set to unique id of the observed participant and the To attribute set to the unique id of a Binding with no Value elements attached.
- A murder-for-hire plot can be represented with a set of Binding elements standing in for roles such as Instigator, Hitman, Middleman, Payoff, Victim, and so on. These bindings will have no Value elements attached. Thus, Binding elements provide a means to represent patterns, and in fact provide ODL with a limited version of the capabilities that are found in a rich pattern expression language such as PatternML (Harrison, 2002).
- As a murder-for-hire is investigated, suspects can be attached to roles as they are discovered, and the strength of the suspicion can be reflected in the probability of that attachment. For example, if in a given suspected murder it appeared that Boris was possibly the instigator and that the return of blackmail photographs was the payoff, Boris could be attached as a Value to the Instigator Binding element with probability .5, and a Resource element representing the blackmail photographs could be attached to the Payoff Binding with, say, probability .75.

- As the murder-for-hire investigation hits complications, multiple suspects might develop. In this case, they can also be attached to the Bindings. For instance, if we uncover the possibility of an alternate plot, we might also attach Ilya to the Instigator Binding with probability .3 and a Resource element representing a large sum of money to the Payoff Binding with probability .2.

Binding elements thus provide a flexible way to represent partially known facts, somewhat abstract patterns, and instantiations of those patterns with specific Nodes attached to roles. Figure 22 is an expanded version of Figure 21 that uses Bindings.

```
<Model>
   <Agents>
      <Binding ID="AH" Name="Hijacker" VariableClass="AGENT"/>
      <Binding ID="AHL" Name="Hijack Leader" VariableClass="AGENT">
         <NodeRef Node="A5" ProbabilityOfNode="0.5"/>
         <NodeRef Node="A10" ProbabilityOfNode="0.3"/>
      </Binding>
      <Agent ID="A5" Name="Akhtar"/>
      <Agent ID="A10" Name="Latif"/>
      <Agent ID="A18" Name="Shakir"/>
   </Agents>
   <KnowledgeItems>
      <Knowledge ID="KHHC" Name="Hand-to-Hand Combat Skill"/>
   </KnowledgeItems>
   <Resources>
      <Resource ID="RBC" Name="Box Cutter"/>
   </Resources>
   <Locations>
      <Location ID="LHF" Name="IC 814">
         <Annotation>
            Hijack Flight
         </Annotation>
      </Location>
   </Locations>
   <Organizations>
      <Binding ID="OTO" Name="Terrorist Organization" VariableClass="ORGANIZATION">
         <NodeRef Node="O14"/>
      </Binding>
      <Binding ID="OTB" Name="Terrorist Cell" VariableClass="ORGANIZATION">
         <NodeRef Node="O3" ProbabilityOfNode="0.1"/>
         <NodeRef Node="O4" ProbabilityOfNode="0.3"/>
      </Binding>
      <Organization ID="O3" Name="Harkat-ul-Ansar Mumbai Cell #1" Network="NHC"/>
      <Organization ID="O4" Name="Harkat-ul-Ansar Mumbai Cell #2"/>
      <Organization ID="O14" Name="Harkat-al-Ansar Mujahadeen"/>
   </Organizations>
   <Networks>
      <Network ID="NHC" Name="HUM Cell Network #1">
         <Nodelist>
            <NodeRef Node="A5"/>
            <NodeRef Node="A10"/>
            <NodeRef Node="A18"/>
         </Nodelist>
         <Edgelist>
```

```
            <Edge Source="A5" Dest="A10" ID="E1"/>
            <Edge Source="A5" Dest="A18" ID="E2"/>
            <Edge Source="A10" Dest="A18" ID="E3"/>
        </Edgelist>
    </Network>
  </Networks>
</Model>
```

Figure 22. Expansion of Figure 1 that uses Binding elements.

*ODL Modularity*

ODL has two somewhat conflicting goals:

1. Provide a common language for all tools to be integrated. This goal argues for a least common denominator approach, that is, for a subset of network organizational representations shared by all current and future integrated applications. All else equal, this common denominator will not change when a new tool is added.
2. Provide a comprehensive means for each tool to represent all the data it requires, whether or not any other tool can use that data. This goal argues for a least common multiple approach, that is, for a large set of representations that cover the needs of all integrated applications. It is likely that any new tool to be integrated will add at least some new requirements to the mix, and therefore the common multiple is likely to change when a new tool is added.

In order to accommodate both goals, ODL was architected in a modular fashion in as the W3C has outlined for XHTML (W3C, 2004).

The first issue is the structure of the schema for ODL itself. The strategy is to make heavy use of attribute and element groups in the definitions so that precision can be exercised in redefining any aspect of ODL. Those redefinitions, of course, need to be limited to extensions, much as in object oriented inheritance. Figure 23 shows a before-and-after snippet of the definition of the element 'Agent.' It's clear that the modular definition is considerably more verbose than the ordinary definition, but it is also far more granular, and this provides more precision to extend the definition with new attributes and constituent elements.

```
<xs:element name="Agent">
  <xs:complexType>
    <xs:complexContent>
      <xs:attribute name="Name" type="xs:token" use="optional"/>
      <xs:attribute name="ID" type="xs:ID" use="required"/>
      <xs:attribute name="NodeProbability" type="Probability" use="optional"/>
      <xs:attribute name="BeginApplicability" type="Date"
        use="optional" default="ALWAYS"/>
      <xs:attribute name="EndApplicability" type="Date"
        use="optional" default="NEVER"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

a. Ordinary Element Definition

```
<xs:attributeGroup name="Node.attlist">
    <xs:attribute name="Name" type="xs:token" use="optional"/>
    <xs:attribute name="ID" type="xs:ID" use="required"/>
    <xs:attribute name="NodeProbability" type="Probability"
          use="optional"/>
    <xs:attribute name="BeginApplicability" type="Date"
          use="optional" default="ALWAYS"/>
    <xs:attribute name="EndApplicability" type="Date"
          use="optional" default="NEVER"/>
</xs:attributeGroup>
<xs:group name="Node.content">
        <xs:sequence/>
</xs:group>
<xs:complexType name="Node.type" abstract="true">
    <xs:group ref="Node.content"/>
    <xs:attributeGroup ref="Node.attlist"/>
</xs:complexType>
<xs:element name="Node" type="Node.type">
    <xs:annotation>
        <xs:documentation>Node with ID</xs:documentation>
    </xs:annotation>
</xs:element>

<xs:attributeGroup name="Agent.attlist"/>
    <xs:group name="Agent.content">
        <xs:choice/>
    </xs:group>
<xs:complexType name="Agent.type">
    <xs:complexContent>
        <xs:extension base="Node.type">
            <xs:group ref="Agent.content"/>
            <xs:attributeGroup ref="Agent.attlist"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:element name="Agent" type="Agent.type"/>
```

b. Modular Element Definition

Figure 23. Ordinary and modular definitions of the Agent element

The second consideration concerns namespaces as ODL is extended. Should extensions have their own namespaces? When other XML Schema-based languages are used as extensions, should they have their own namespaces? Table 4 and Table 5 outline some of the tradeoffs, and the approach selected for ODL is outlined in bold. In these tables, AsamML is an example of a direct extension to ODL, and XMLBIF (which is assumed to have a prefix "bif:") is an example of a existing language to be incorporated into an extension. The ODL namespace has a prefix of "odl:" when it has one at all.

Table 4. Namespace considerations for core+extensions architecture for ODL.

| Option | For example… | Advantages | Disadvantages |
|---|---|---|---|
| Separate namespaces | <bif:NETWORK><br><odl:Binding><br><HMM> | • Simplicity of specifying the schema<br>• Guarantee of no name collisions | • Some inconvenience for users, especially when foreign-namespace elements or attributes have been attached to default namespace elements<br>• Not particularly in line with XHTML-style modularity approach |
| Hybrid – some within one namespace, some separate namespaces | <NETWORK><br><odl:Binding><br><HMM><br><br>--or—<br><br><bif:NETWORK><br><Binding><br><HMM> | • Maximum flexibility<br>• Very consistent with W3C Modularization approach if structured appropriately | • Finding the right mix might be tricky<br>• If not done with some consistent rules, could be even more confusing to users than separate namespaces<br>• Pragmatics require some design |
| One namespace | <NETWORK><br><Binding><br><HMM> | • Simplicity for the user | • Possible name collisions<br>• Difficult pragmatics<br>• Module developers must be cognizant of other potential modules |

Table 5. Namespace options for hybrid solution to core+extensions architecture for ODL.

| Option | For example… | Advantages | Disadvantages |
|---|---|---|---|
| ASAM is redefined ODL + intact XMLBIF in its own namespace | <Model><br><bif:BIF><br><bif:NETWORK><br></bif:NETWORK><br></bif:BIF><br>…<br><HMM … /><br></Model> | • All models to integrate are centered around ODL concepts<br>• It's clear how to proceed when two or more extensions<br>• Allows alternate BN (and other) representations | • Requires discipline only to extend (not restrict) ODL<br>• Inflexible |

| | | | |
|---|---|---|---|
| ASAM is redefined ODL + lightly modified XMLBIF in its own namespace | <Model><br><bif:BIF><br><bif:NETWORK HMM="HMMOO1"><br></bif:NETWORK><br><bif:BIF><br>…<br><HMM…/><br></Model> | • All models to integrate are centered around ODL concepts<br>• It's clear how to proceed when two or more extensions<br>• Allows alternate BN (and other) representations | • Requires discipline only to extend (not restrict) ODL<br>• Slight increase in implementation difficulty<br>• May require minor refactoring of non-ODL modules |
| ASAM is redefined XMLBIF + intact ODL | <BIF><br>…<br><odl:Organization><br>…<br></odl:Organization><br>…<br><NETWORK><br><HMM/><br></NETWORK><br></BIF> | • Preserves ODL core unchanged<br>• Extracting ODL-only info is easy. | • Models are centered around XMLBIF. What about models involving other extensions to ODL?<br>• Inflexible |
| ASAM is redefined XMLBIF + lightly modified ODL | <BIF><br>…<br><odl:Organization DBN="DBN001"><br>…<br></odl:Organization><br>…<br><NETWORK><br><HMM/><br></NETWORK><br></BIF> | • Mostly preserves ODL core | • Models are centered on XMLBIF. What about models involving other extensions to ODL?<br>• Slight increase in implementation difficulty<br>• Requires that non-ODL modules be refactored according to embedding guidelines |

An example of how all this works is AsamML. AsamML is an ODL-based language used by an organizational analysis and monitoring tool named ASAM. In addition to the normal ODL constructs, AsamML makes provisions for hidden Markov model (HMM) specifications and dynamic Bayesian networks, neither of which is accommodated in core ODL.

According to the logic just discussed, AsamML is in the same namespace as ODL, since it is an extension. AsamML internally makes provisions for the specification of HMMs. AsamML uses another language to represent the Bayesian network information, namely XMLBIF (Cozman, 1998).

Because XMLBIF is a standalone language, it will get its own namespace in AsamML. But for optimal usage, XMLBIF itself will need minor adjustments, namely to create linkages from the

Bayesian networks back to the HMMs in AsamML. XMLBIF is therefore extended in a language called XMLBIF-ASAM, in the XMLBIF namespace. XMLBIF-ASAM is then imported into AsamML to complete the picture

Figure 24 shows the situation graphically.

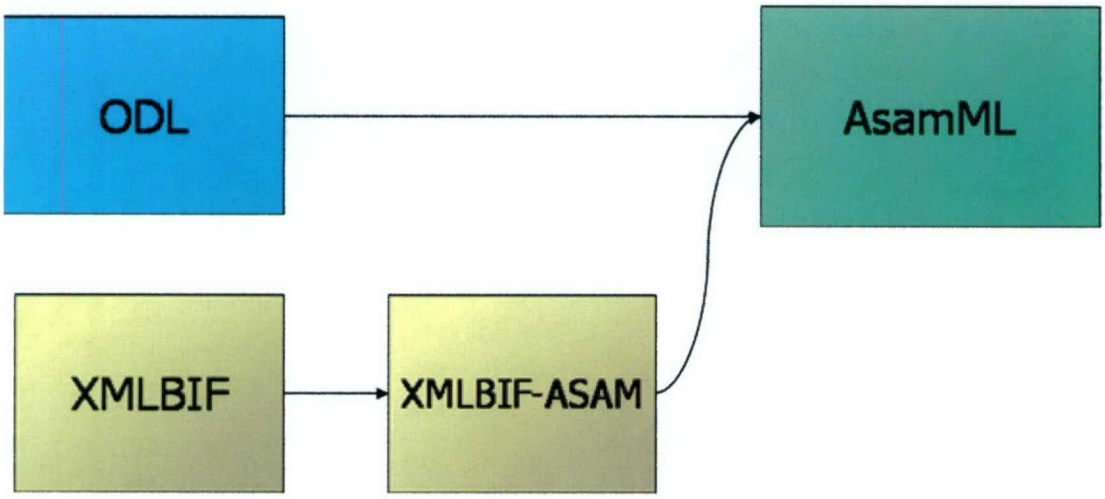



Figure 24. Component languages of AsamML.

*Conclusion*

This brief overview has sketched the basics of ODL. Many additional details are available in Appendix B.

### TIE with Cycorp

Aptima, Cycorp, and CMU have an ongoing project to use existing NEMESIS bus-oriented integration infrastructure, and notably ODL, to create the opportunity to extract and transform information from a large ontology of facts about terrorism, called Terrorism Knowledge Base (TKB), and analyze it with a network organizational analysis tool named Organization Risk Analyser (ORA). We made substantial progress this year, in that we were able to show an end-to-end integration from TKB through ODL to ORA and its native language DyNetML.

The chief difficulty that confronted us was that TKB, in its native form, stores a very wide variety of facts and can deduce from them a larger number of which only selected portions of them can be used by ORA. ORA, for its part, expects to encounter data describing dynamic networked organizations. While this data is available in TKB, finding it and transforming it proved to be an interesting project.

Our method for the integration was to export selected knowledge from TKB into Web Ontology Language (OWL), to transform the OWL export into ODL, to transform the ODL into ORA's native language, DyNetML, and then to use ORA on the resulting data. The step of transforming OWL to ODL rather than directly to DyNetML enabled the extracted knowledge to participate in

NEMESIS' bus-oriented integration, and therefore to integrate with other analytic tools that are or will be integrated with NEMESIS.

*Cyc KB and TKB*

NEMESIS, ODL, and ORA have already been described, but it is worth taking a moment to describe Cycorp's Terrorism Knowledge Base (TKB) from which terrorist network information was extracted. For the past 18 months Cycorp, Inc. has been building a resource designed to fundamentally transform the way intelligence analysis is done: TKB. The vision for TKB is ultimately to contain all relevant knowledge (beginning with unclassified knowledge) about terrorist groups, their members, leaders, ideology, founders, sponsors, affiliations, facilities, locations, finances, capabilities, intentions, behaviors, tactics, and full descriptions of specific terrorist events. Led by world-class experts in terrorism, knowledge enterers have been building the TKB at a rate of up to 100 assertions per hour. The knowledge entry tools supplied by Cycorp are simple enough for unassisted domain experts to use. However, the knowledge entered is not represented as simple database entries, but rather is stored as statements in mathematical logic, suitable for computer understanding and reasoning. In fact, the TKB has a growing complement of reasoning modules for integrating data and correlating observations, generating scenarios, answering questions and composing explanations. The TKB, together with its reasoning modules, will be a fully computer-understandable terrorist knowledge base capable of supporting computer reasoning to aid analysts.

The TKB is an augmentation of the existing Cyc Knowledge Base (Cyc KB), which has been under intensive construction for the past 20 years. The Cyc KB contains a formalized representation of large tracts of consensus reality, encoded in tens of thousands of terms and millions of hand-entered assertions, organized into hundreds of contexts (called "microtheories"). Most of the current content of the Cyc KB consists of general facts about all sorts of everyday objects and activities. It also contains "almanac-style" facts about individual countries, ethnic groups and organizations. Prior to launching the development of the TKB, the Cyc KB already had a substantial amount of knowledge relevant to terrorist activity, such as knowledge about geopolitical events, WMD, and military hardware.

The TKB effort has so far added to the Cyc KB knowledge of over 2000 individual terrorists, over 700 different terrorist groups and over 6500 terrorist attacks. The representations of these individuals, groups and events are involved in over 200,000 assertions such as "Lashkar-e-Taiba was founded in 1990" and "The October 28th, 2002 terrorist attack in Amman, Jordan killed a United States diplomat". Every fact represented thus far in the TKB has been culled from open source data.

*Extraction, Transformation, and Analysis*

Here are the steps we followed to achieve the integration of TKB and ORA via NEMESIS' bus-oriented integration capabilities. The effort describe here is a first step; we deliberately chose to work with a small subset of knowledge from TKB in order to avoid trying to address too many issues (like scalability) at once. Still, the fact that we were able to achieve an end-to-end integration illustrates the feasibility of the bus-oriented approach, and points to an exciting future for this effort.

Export from TKB to OWL

TKB first produced an OWL export of content that consisted of a small number of persons, terrorist events, and terrorist organizations and a sampling of the various relations and attributes involving those individuals. Initially restricting the size of the export allowed us to study which of the widely varied TKB data types were best suited for use by NEMESIS. Based on feedback from this initial study we took into consideration the sort of links that ORA could best utilize when we produced the next export.

For the second export we started with a list of 373 individuals that were mentioned in Marc Sageman's Matrix database of social network data on the 9/11 conspirators. That database had been mapped into the TKB earlier, thus integrating it with information on those individuals already present in the system. The subject matter experts (SMEs) already entered a significant number of assertions about most of these individuals. Each of the individuals in the Matrix database was involved in between 50 and several hundred assertions. Since links between individuals were of prime importance in doing dynamic network analysis, we decided to concentrate on various types of "personal association relations" that involved the individuals mentioned in the Matrix. Personal association relations are binary relations that relate instances of the class of persons. Examples (written in Cyc's native representation language CycL) include #$religiousTeacherOf, #$businessPartners, #$subordinates, etc. The java extraction program gathered these identified assertions and grouped the contained terms by type: predicate, collection and individual. OWL is an XML syntax for describing and transmitting ontological information, given certain expressiveness limitations (e.g. not first order predicate calculus).

This is an example of a CycL assertion:

```
(boss Terrorist-Salim OsamaBinLaden)
```

For each Sageman individual the identified assertions were converted into OWL format. The extracted OWL version of the example assertion looks like this:

```
<AdultMaleHuman rdf:ID="Terrorist-
    Salim">
  <rdfs:label xml:lang="en">
    Mamdouh Mahmud Salim
  </rdfs:label>
  <guid>
    dff74888-a901-41d8-9051-
        ea6e5432b01a
  </guid>
  <boss rdf:resource="#OsamaBinLaden"/>
</AdultMaleHuman>
```

Transforming the OWL to ODL

The resulting file contained 2,926 lines of OWL, which were transformed to ODL as follows.

First, for ease of translation and because there were no namespace conflicts, namespaces were removed from the OWL export. Though the result could no longer be considered OWL, there was no loss of information in the resulting XML file.

Next, again for ease of translation, an XML schema that fully covered the data was created from the XML file. This is a standard function in XML editors such as Altova's XMLSpy (Altova, 2005.) The step is useful because the OWL export has considerably more freedom than a constrained language such as ODL, and the schema provided an inventory of OWL types that needed translation. The resulting schema yielded a list of all element and attribute types that needed to be translated into ODL.

With schema in hand, the next step was to write XSLT to transform the file to ODL. Each element in the export file was mapped into one of ODL's node types, and all information was preserved in the transformation. Relational attributes on elements in the original file, such as 'friends,' 'significantEventAcquaintance,' and 'subordinates,' turned into values of the Color attribute on Edges.

The process resulted in 4,775 lines of ODL. ODL corresponding to the OWL extract in the last section looks as follows.

```
<Agent ID="Terrorist-Salim"
    Name="Mamdouh Mahmud Salim">
  <Annotation>
    TKB type: AdultMaleHuman
    guid: dff74888-a901-41d8-9051-ea6e5432b01a
  </Annotation>
</Agent>

<Edge ID="boss95884976" Color="boss"
    Source="Terrorist-Salim"
    Dest="OsamaBinLaden"/>
```

Transforming the ODL to DyNetML

Since NEMESIS is already integrated with ORA, transforming ODL to DyNetML is straightforward. The process consists of a chain of XSLT transformations, the most important of which sorts networks by node types.

The only major change to the existing process was that the direction of the edges inherent in the exported information sometimes needed attention. For instance, in some cases, there was an edge from an Organization node to an Agent node labeled 'hasMember,' and in other cases there was an edge from an Agent to an Organization, such as 'isLeaderOf.' In these cases, the direction of the edge was forced to be uniform; the label/color of the edge can be changed to make this sensible. In the present example, the 'hasMember' edge became a 'memberOf' edge. No information was lost in this process.

Translation resulted in 6,117 lines of DyNetML. DyNetML corresponding to the OWL and ODL from the examples above looks like this.

```
<node id="Terrorist-Salim"
    title="Mamdouh Mahmud Salim"/>
<edge source="Terrorist-Salim"
    target="OsamaBinLaden" type="double"
    value="1">
  <properties>
    <property name="color" type="string"
        value="boss"/>
  </properties>
</edge>
```

Using the DyNetML in ORA

The resulting DyNetML was loaded into ORA. The results were limited—by design for the first iteration—by the fact that the data set used is a small sample of the available open-source data on al Qaeda and by the fact that the data as provided did not discriminate by time (e.g., operatives who died in year 1 and those who joined in year 3 appear in the same graph). The resulting network and findings are therefore illustrative but not necessarily indicative of the nature of al Qaeda. The data set is comprised of 574 human actors, 150 events, 22 locations and 19 organizations. No data was provided on knowledge or resources, though we expect it will be available in future iterations, at which point it will be possible to identify the emergent leaders in the group using ORA.

There were 6 completely separate sub-groups. This is an expected artifact of the missing data in the small initial extraction. With more complete data in future iterations, key features of ORA will be used, though they couldn't be used with this data set. These include identification of emergent leaders, characterizing cognitive or resource differences among actors, and locating agents who have special expertise.

After loading the DyNetML into ORA, we dropped all nodes that were not connected to other nodes. We then visualized the resulting network (see Figure 25). In this figure we see the 6 components (3 of which represent small cells) involving actors, events, and groups. The largest cluster centers around al Qaeda and contains Osama Bin Ladin. There is little connection between actors and events due to the fact that the underlying data analyzed contained membership and sponsorship but little information on who was involved in what events. Application of the grouping routines in ORA would further break the large cluster into 4 groups, which could then be further divided. We note that for most real data that CMU has examined the groups are not as "visible" as in this sample of data.

ORA analysis of the data reveals the key actors. Those who stand out as being "in the know" (i.e., have a high degree of centrality) are Osama bin Laden, Sheikh Mohammed, Mohamed Atta, and Ayman al-Zawahiri. Those who stand out as "connecting disconnected groups" (high in betweeness and low in degree centrality) include Abu Zubaydah and Ramzi Mohammad Abdullah bin al-Shibh.

From an effects based operations perspective you might want to utilize those in the first group for information and those in the second for passing rumors.

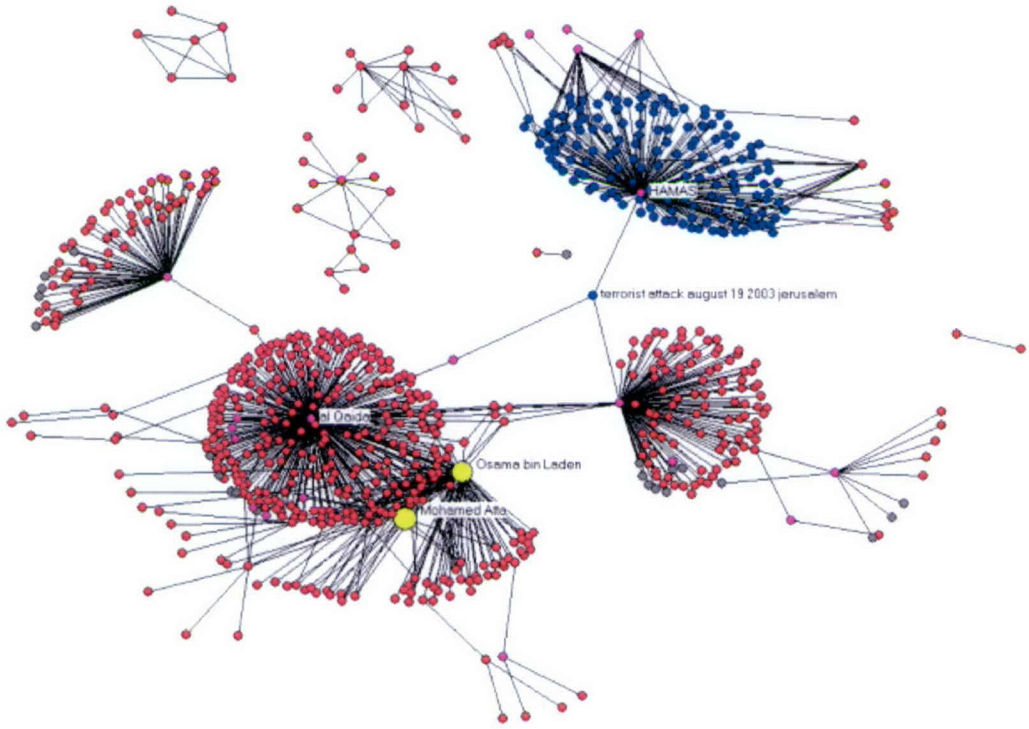



Figure 26. Overall meta-network sample visualization data

As part of the drill down capability the analyst can explore the sphere of influence around one or more actors. The sphere of influence is termed the "ego net," consisting of the set of other nodes the actor or group of interest is connected to and the connections among them. In Figure 27, the sphere that surrounds Atta and bin Ladin is shown. Here we see that bin Ladin is connected to almost everyone that Atta is connected to and that, in addition, Bin Ladin is connected to a large group of others among whom there appear to be few connections. If this data is correct, it would suggest that it would be difficult to influence bin Ladin (because he receives information from many separate sources) and easier to influence Atta.

The ORA context report notes how the values for the group terrorist network being analyzed compares with a random network and other known networks. Here we find that the overall network is very sparse (density = .0008) as compared to other networks CMU has examined that have densities around .024. This, in conjunction with the fact that most clusters are centered on a few nodes (note that they look like exploding fireworks) suggests that the data available is highly incomplete and additional data is needed.

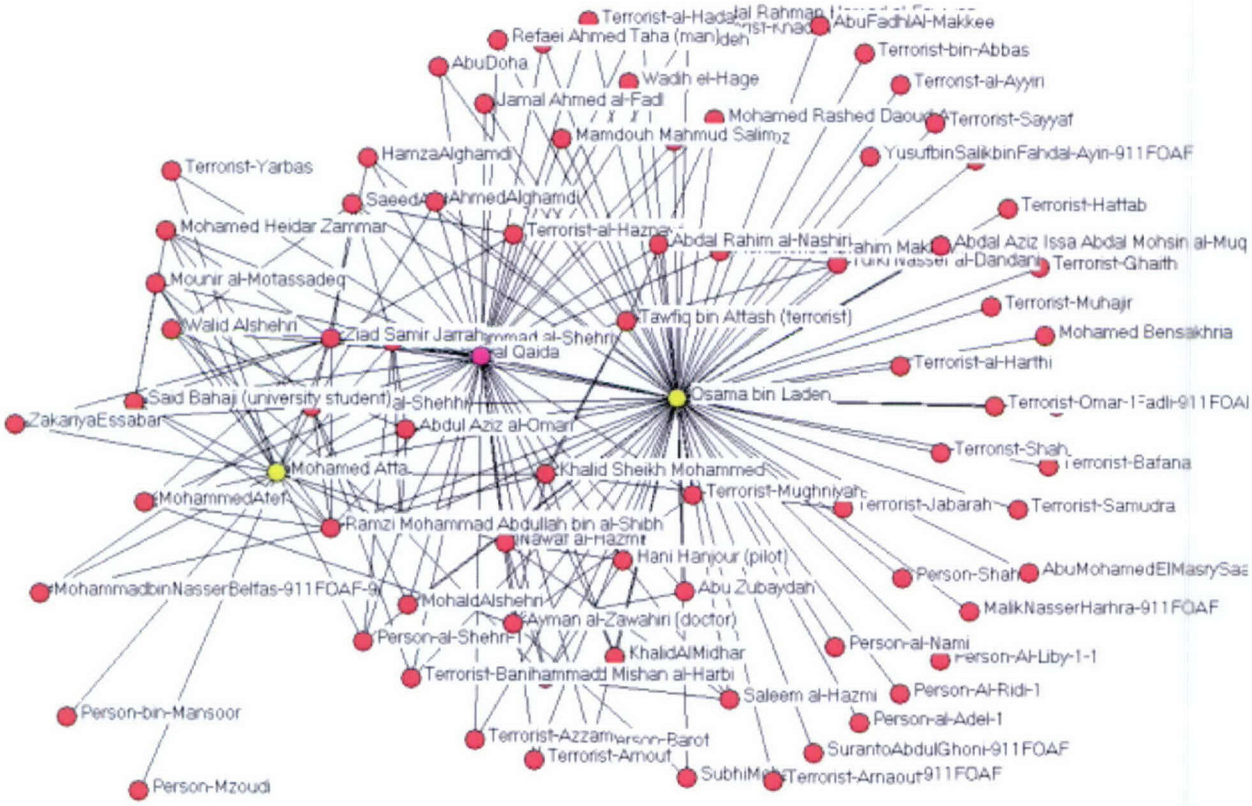



Figure 27. Sphere of influence around Atta and bin Laden

## Conclusion and Future Steps

The work described in this paper represents an early successful end-to-end integration of TKB and ORA via ODL and the NEMESIS bus-oriented integration infrastructure. It was not necessary to change TKB, ODL, ORA, or any other application. All that was necessary—besides the effort involved in thinking through what information in TKB to extract—was a small amount of new extraction and transformation code. Further, because the integration was performed via the bus-oriented architecture, the extracted and transformed data will be available to any other tool that also integrates with the bus-oriented architecture.

It is clear that integrating the knowledge in TKB with the analysis power of a tool like ORA provides substantial benefit. We have shown that this can be done with the bus-oriented integration provided by NEMESIS and ODL. This is also a way to make many other integrations possible with minimum effort. As additional applications and data sources are integrated, the utility of the resulting integrated system will grow faster than linearly with their number, assuming that each new application adds value to more than one other component.

We plan the following as logical next steps:

- We will add **additional data types** to the extraction and transformation process, starting with information about what knowledge the people and organizations have.
- We will **scale up the amount of data** extracted and transformed from TKB. In the early phases of the integration, we felt it wise to limit the size of the data set in order to ensure that we understood the necessary steps in the extraction and transformation. We are now in a position to work with considerably more data. This step should remove some of the data sampling limitations that became visible in ORA.
- We will add **additional network organizational analysis** tools to the integration. The point of bus-oriented integration, after all, is the ease with which new tools can be integrated.
- We will automate the transformation in the reverse direction, **from ODL to TKB**.
- We will **expand ORA** to handle additional entity types available in TKB.
- We will use Cyc's inferencing capabilities to **optimize the structure of the exports**.

Our hope is that this effort paves the way for additional integration efforts with additional data sources, knowledge bases, and analysis tools. Progress towards combining the powerful but diverse capabilities already available will go far towards putting the right information in analysts' hands at the right

## *Task 4. Support for Collaboration on Network Organizational Models*

NEMESIS presents an unusual collaboration context because collaborations will likely take place over an extended period of time. The work products of the collaboration, ODL descriptions of organizations, transactions, and documents associated with the organizations, will also evolve over time. In some cases, there will be multiple versions of those work products, some exploring speculative "what if" kinds of scenarios, others describing the team's evolving understanding of the situation in a sequence of revisions and versions. In addition, there is a strong need to understand the context of each document—who is the author, what sources were used, when was it first produced, who modified it last (and when and why), what documents need to be revised given that there is a new version of another document, and so on. NEMESIS is developing a capability called *collaborative versioning* to address these issues.

### Collaborative Versioning

Simple collaborations via email, instant messages, chat, and discussion forums generally have no need for revisions. They can be saved and restored via simple archiving. As the collaborations grow more sophisticated, though, the resulting artifacts of collaboration grow in complexity, and increasingly can benefit from an approach to archiving that tracks versions and configurations of them.

For intelligence analysts served by NEMESIS, examples of these artifacts would be report series, daily intelligence summaries, status updates, scenarios and models of terrorist organizations, and structured argumentation of various kinds. Broader examples of these artifacts are collaboratively authored documents, strategies and plans, and "what-if" analyses.

New versions and revisions of these artifacts come from several sources. First, there may be new data or new interpretations of existing data to be incorporated in an analysis. Second, there may be multiple

hypotheses about the actual state of the world, some of which are the product of "what if" exploratory analysis, all of which need to be managed. Finally, a document may be based on other documents (for instance, it might summarize them), and those documents might change, resulting in the need to change the dependent (summarizing) document. Figure 28 illustrates example dependencies among documents.

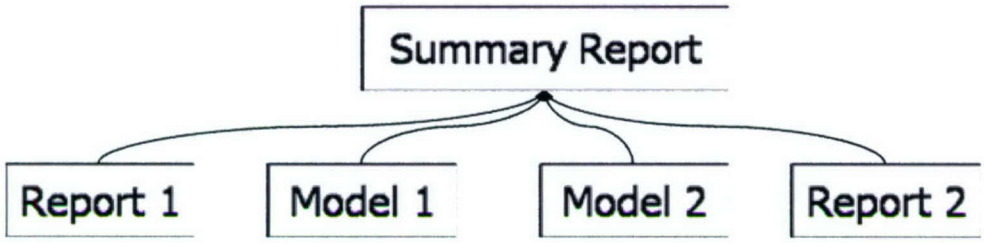



Figure 28. Sample document dependencies.

Figure 29 illustrates an example set of collaborative documents under version control. There is a mainline, or "trunk" line of development, here indicated by the sequence $C_0, \ldots, C_n$. This represents an increasingly elaborated and updated sequence of documents or models (which may themselves represent a sequence of documents or models.) The tip of this sequence, $C_n$, is the most recent version, and represented the collective best guess about the state of the world. Variants of this sequence, called *branches*, occur when an out-of-the-mainstream idea, say a "what if" analysis, is performed, or when extended development of a line of thought otherwise requires isolation from the activity in the mainstream. If and when collaborative participants decide to integrate their work back into the main line of thought, the operation is called a *merge*.

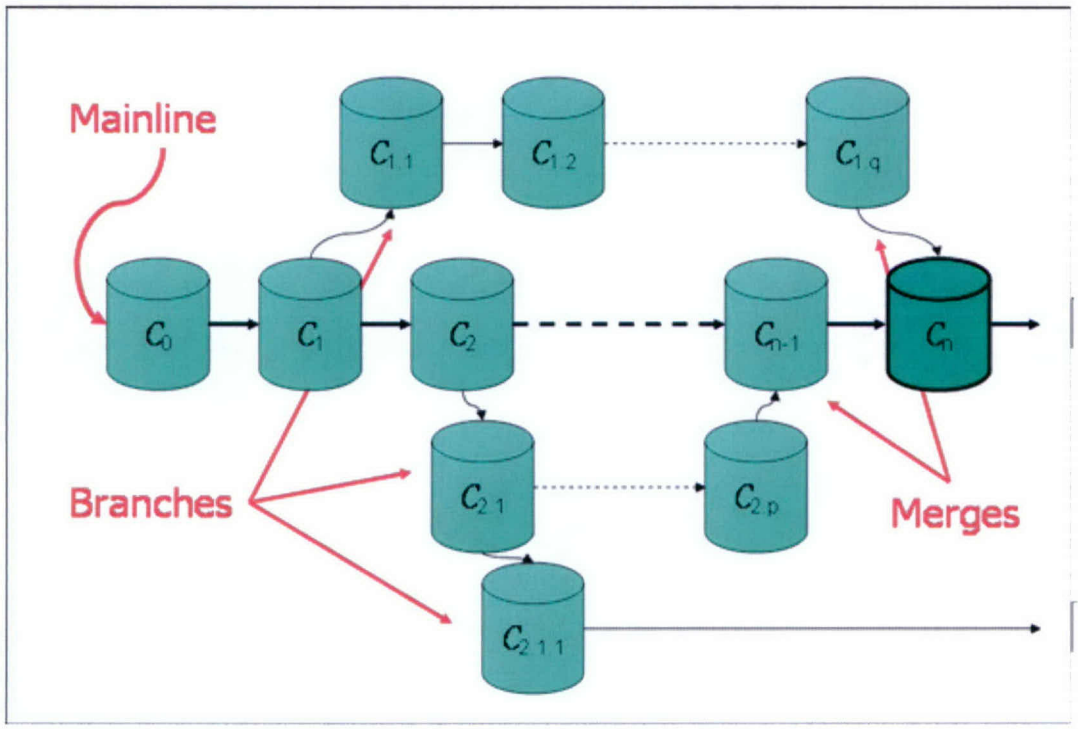



Figure 29. Example version management for collaborative versioning.

We call a system that will manage collaboratively produced, revisable artifacts like this a *collaborative versioning* system. It provides several important benefits:

- **Protection from inadvertent or unwanted changes**. Authors who make changes to a document that they later decide to discard may have trouble reconstructing the earlier version unless they are using collaborative versioning.
- **Capture and search of metadata**. Metadata describing documents and changes to those documents are automatically labeled by author and by the time of change and by an application-specific form to capture additional metadata from the author at the time the change is made. The metadata can subsequently be searched to find those documents and versions meeting certain criteria, such as the changes made by a given author or the version of a document as of last Tuesday.
- **Reconstruction of the collaboration**. Collaborative versioning provides a convenient means to reconstruct the state of the collaboration as of a given date or event. In addition, the metadata accompanying revisions can be queried to identify those artifacts and changes sharing certain properties, such as having the same author or source document.
- **Understanding revisions**. With collaborative versioning, it is simple to retrieve an author's notes about why a certain change was made, and to view the differences between any two versions of a document.
- **Concurrency control**. When two or more authors work on the same document at the same time without concurrency control, one of them could lose their work, often depending on who saves their work last. Collaborative versioning solves this problem. Older versioning systems used a checkout-modify-checkin, or *pessimistic*, scheme for preventing simultaneous conflicting changes. Under this system, only one participant at a time can make changes to a document. This is effective for its intended purpose, but totally blocks multiple participants from working on the same document at the same time, even if they are making changes to different and unrelated portions of the document. The more modern dopy-modify-merge, or *optimistic,* control allows an unlimited number of people to make changes to a file and detects incompatible changes, called *conflicts*, when the participants attempt to commit their edits into the repository. This provides a great deal of convenience when the risk of conflict is low, as is the case for collaboration. Hence, optimistic control is ideal for collaboration.
- **"What-if" and exploratory collaborations**. With collaborative versioning, it is possible to isolate exploratory changes until they are ready to be merged back into the main collaboration. And exploratory branches may themselves have exploratory branches.
- **Synchronization** of artifacts in a collection of documents. Sometimes a document depends on other documents: a book may depend on a set of chapters, a report may depend on a set of source documents. When this is true, it is important to be able to quickly find source documents newer than the top-level document and to control (and to be able to reconstruct) which versions of the source documents went into a given version of the top-level document. Collaborative versioning provides this capability.

Collaborative versioning functionality is similar to that found in source code management (SCM) systems for software development, but the areas of emphasis are somewhat different. Table 6. Comparison of version control requirements for software
development and for collaboration. outlines several such differences. Most notable are the

observations that concurrency control is more important for software development than for collaboration (because concurrency conflicts are more likely in software development than in typical collaboration) and that flexibility in reconstruction and search with respect to metadata are more important in collaboration (because participants are more likely to care about the state of the collaboration at a specific date, or the contributions of a specific author, or, for the intelligence community, the changes that came about because of a given report or due to a given source).

Table 6. Comparison of version control requirements for software development and for collaboration.

| Feature | Software Development | Collaboration |
|---|---|---|
| Revision **Synchronization** | Major | Major |
| **Protection** from Inadvertent Changes | Major | Major |
| Concurrency **Control** | Major | Minor |
| Configuration Management and **Reconstruction** | Minor | Major |
| Metadata and Change **Tracking** | Minor | Major |
| Differencing and Merging | Character/line oriented | Semantic |

In addition to these requirements, there are several additional desiderata for collaborative versioning.

- **Transparency.** The ideal for version control systems is that they are almost invisible unless needed. This is because their operation can be quite confusing for the uninitiated and more generally because if they are perceived to interfere with an analyst's normal workflow, they run the risk of simply being ignored.
- **Comprehensibility.** The system should offer insight rather than confusion in explaining the relationships among documents and their versions in the collaboration.
- **Scalability.** It should be easy to add new types of documents and models to the system, and human nor system performance should suffer as the number of documents under version control grows large.
- **Compatibility.** The system should operate in a wide variety of contexts ranging from closed networks to collaborative platforms such as Vignette Collaboration Server.

NEMESIS collaborative versioning functionality builds upon an open source version control system called Subversion (SVN) that provides a solid foundation for meeting the listed requirements. SVN carries into the future the functionality of the well-known CVS version control system. SVN, like CVS, works on a copy-modify-merge (optimistic) scheme, the preferred solution for collaborative versioning. Relative to CVS, SVN has additional, modern features, as spelled out on its web site (http://subversion.tigris.org):

- **Directories, renames, and file meta-data are versioned.** Lack of these features is one of the most common complaints against CVS. SVN versions not only file contents and file existence,

but also directories, copies, and renames. It also allows arbitrary metadata ("properties") to be versioned along with any file or directory, and provides a mechanism for versioning the `execute' permission flag on files.

- **Commits are truly atomic.** No part of a commit takes effect until the entire commit has succeeded. Revision numbers are per-commit, not per-file; log messages are attached to the revision, not stored redundantly as in CVS.
- **Apache network server option, with WebDAV/DeltaV protocol.** SVN can use the HTTP-based WebDAV/DeltaV protocol for network communications, and the Apache web server to provide repository-side network service. This gives SVN an advantage over CVS in interoperability, and provides various key features for free: authentication, path-based authorization, wire compression, and basic repository browsing.
- **Standalone server option.** SVN also offers a standalone server option using a custom protocol (not everyone wants to run Apache 2.x). The standalone server can run as an inetd service, or in daemon mode, and offers basic authentication and authorization. It can also be tunneled over ssh.
- **Branching and tagging are cheap (constant time) operations**. There is no reason for these operations to be expensive, so they aren't. Branches and tags are both implemented in terms of an underlying "copy" operation. A copy takes up a small, constant amount of space. Any copy is a tag; and if you start committing on a copy, then it's a branch as well. (This does away with CVS's "branch-point tagging", by removing the distinction that made branch-point tags necessary in the first place.)
- **Natively client/server, layered library design**. SVN is designed to be client/server from the beginning; thus avoiding some of the maintenance problems which have plagued CVS. The code is structured as a set of modules with well-defined interfaces, designed to be called by other applications.
- **Client/server protocol sends diffs in both directions**. The network protocol uses bandwidth efficiently by transmitting diffs in both directions whenever possible (CVS sends diffs from server to client, but not client to server).
- **Performance costs are proportional to change size, not data size**. In general, the time required for an SVN operation is proportional to the size of the *changes* resulting from that operation, not to the absolute size of the project in which the changes are taking place. This is a property of the SVN repository model.
- **Efficient handling of binary files**. SVN is equally efficient on binary as on text files, because it uses a binary diffing algorithm to transmit and store successive revisions

### Collaborative Versioning and Workflow

Our initial target for enhancing the year 1 collaborative versioning system was to write a Groove-based client for SVN. Unfortunately, at around the same time, Groove, Inc. decided to discourage vendors and partners from creating new custom tools for the Groove platform. Other means of Groove integration, namely Web Services and Groove Forms, were available, but upon weighing the benefits and costs, Aptima decided to refocus our efforts. The result was an approach we call Collaborative Versioning and Workflow (CVW), which coordinate collaborative versioning with the local workflow of analysts.

The objectives in this enhancement were to make collaborative versioning overhead as small as possible while at the same time providing additional benefit to the document authors, whether they

be intelligence analysts or generic proposal writers. The key insight was that meaningful versions are created when a task is complete—whether it is the completion of a draft for review, the review of the draft itself, or the release and publication of a final document.

One trouble with implementing something like this is that authors' task lists are not generally available unless the authors are using a workflow system. But generic workflow systems are too rigid and formal for the kinds of missions commonly performed by counterterrorism analysts. They will often deal with *ad hoc* teams of experts working on somewhat new tasks, and in the time it would take to draw a workflow diagram with suitable precision, the task might well be accomplished. In short, a workflow system that will be useful in the collaborative situations envisioned in the Topsail program will need to cover a spectrum of tasks, especially those on the right hand side in Figure 30.

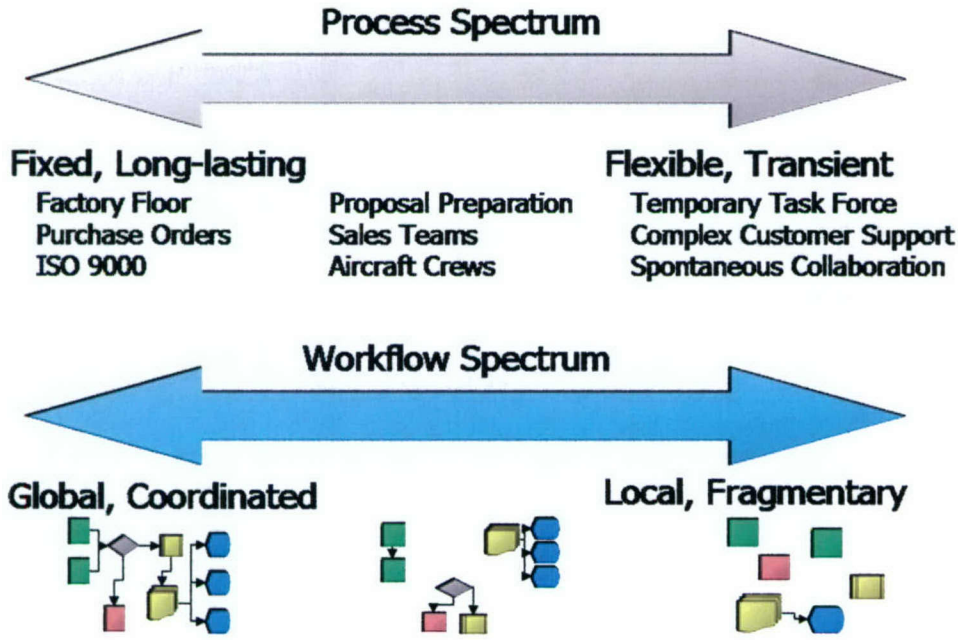



Figure 30. Process and Workflow Spectra

Because a collaborative versioning and workflow system might be of interest outside the intelligence community, Aptima decided to invest in a small IR&D project to develop an initial prototype. As a first step towards a CVW system that combines the local workflows associated with task-oriented collaboration with versioning and configuration management services for the output of those tasks, the on the IR&D project we built a prototype to illustrate some of the functionality of the system. To provide a concrete focus fir the prototype, our target use cases for this prototype were in the area of multi-author proposal preparation.

The basic idea is that meaningful versions of collaboration artifacts are created when tasks or subtasks on a user's local task list are completed. While working on a proposal, for instance, a user will want to create a version when they have a draft ready for a teammate's review or when they have finished incorporating a reviewer's comments. We'll use the term *task point* to refer to the documents at these meaningful versions.

Another set of important states of the document are called *revisions*. A revision is a version of a document that is created for reasons other than the completion of a task—maybe simply to save it to have a backup of the current document state, maybe because the author is going to lunch, maybe because there is some chance the author will want to roll back their changes. Revisions are not yet task points.

In addition to the initial creation of a CVW directory, the CVW prototype will have to major areas of functionality: the creation and management of local task lists, and the document versioning system that will provide standard version control and configuration management services for the documents that result from the tasks.

### Installation

Installation generally follows the same scheme as TortoiseSVN, described in the TortoiseSVN document. One difference is that, unlike TortoiseSVN, there is an opportunity to supply the URL of known repositories (so that they can be presented as options when the user creates a CVW directory.)

Users and their authentication are handled by the standard SVN mechanisms in the prototype. However, in the prototype, a user needs to authenticate themselves only once per CVW directory—the authentication is "sticky."

### Initial Creation of CVW Directory

To start a CVW project, the user identifies a directory (or creates a new directory) for it. The directory may or may not contain documents and/or subdirectories, recursively. The user then right-clicks with the directory selected in Windows Explorer to bring up a context menu. In addition to the standard Windows options on the menu, an additional menu item appears with an option to start a CVW project. If the user selects this option, the user provides the URL of a suitable SVN repository (possibly by selecting one from the menu of known repositories)and the name of the project, and a corresponding directory is created there. If there are documents and/or subdirectories, they are submitted to the repository with comment "Initial revision - <project name>" as well as the identity of the submitter and a date stamp.

CVW directories and subdirectories, as well as the documents that populate them, are identified with a set of icons overlaid on the file icons. As a first pass, these icons are the same as those used by TortoiseSVN, as follows:

 - Nothing has changed since the last synchronization with the repository. No action required.

 - Something has been modified in this file or directory since the last synchronization with the repository. Checkin needed.

 - This document has a conflict with one in the repository.

 - This file or directory is scheduled to be added to the repository.

 - This file or directory is scheduled to be deleted from the repository.

These icons propagate up to the directories that contain them, until the top-level CVW directory is reached.

### Task Lists

Once a CVW directory has been created, the context menu for that directory contains an item dealing with task lists. If no task list exists, there is an item for creating a task list. Selecting it brings up a dialog box. The dialog box is the same for creating the list and for editing the list, and it will be described below.

If a task list already exists, there is an item on the context menu for the directory for editing the task list. Selecting this item brings up the create/edit task list dialog box.

The dialog box for tasks will approximately follow Figure 31 (it will benefit from professional interaction design, but the functionality should remain about the same.)

The user is presented with a number of pre-built task choices, and will build a task list either from them or from task descriptions that are directly entered. Task descriptions may have variables (indicated by '$$') that are placeholders for one or more colleagues, one or more chapters, or other values. In the prototype, the only variables are people to notify, so the wizard actually has a separate screen to specify them. Extensions to other variables will be straightforward.

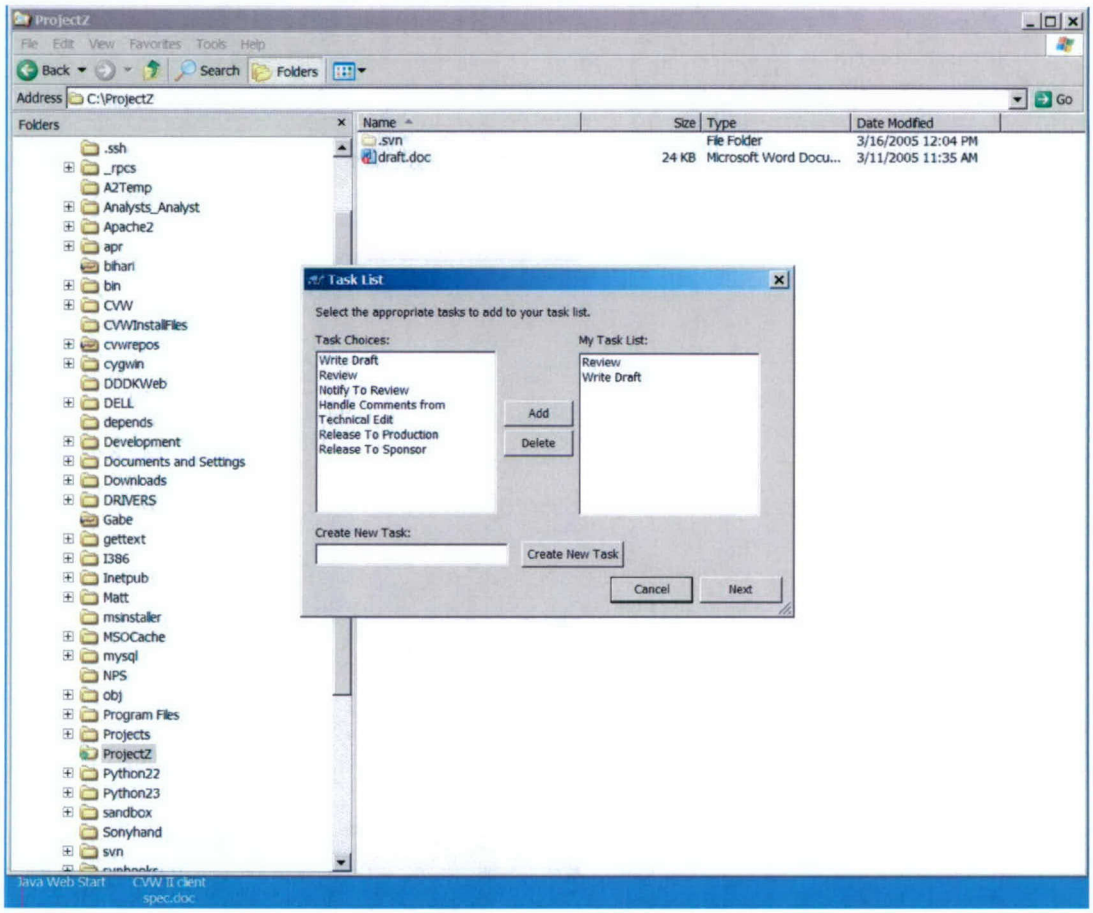



Figure 31. Task List Creation and Edit Screen.

## Collaborative Versioning in CVW

In what follows, the term *CVW entity* refers to a selected document, a selected set of documents, a selected subdirectory of documents, or a mixed selection of documents and document subdirectories. For the purposes of the prototype, the terms document and file are interchangeable, though this might not be true in future versions of the tool.

The services offered by the CVW prototype are "standard" in the sense that they are a subset of those offered by Subversion. Figure 32 shows the CVW Windows Explorer context menu for an entity under CVW control. As you can see, two of the most commonly used options (Update and Intermediate Revision) appear directly on the menu, and less frequently used options are on the CVW submenu.

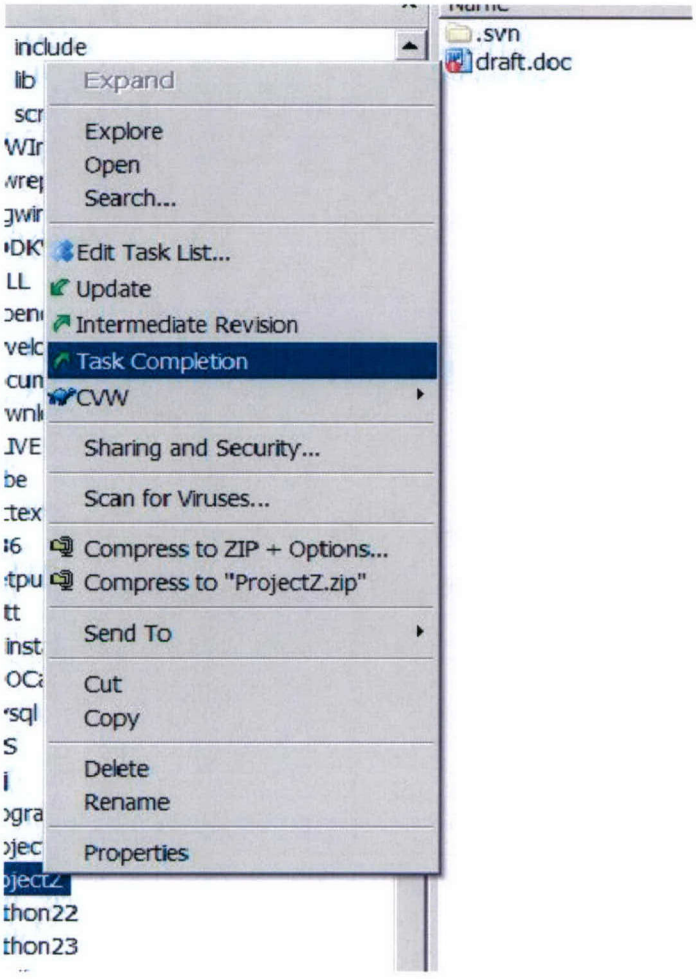



Figure 32. Windows Explorer context menu for a file under CVW control.

'Intermediate Revision' commits the current local copy of the CVW entity to the repository with a comment 'Intermediate Revision.'

Following these two commonly used items, there is an item called 'Task Completion…'. Choosing this item brings up a Task completion dialog box containing a list of Tasks on the left, an textbox for comments on the right, and three buttons, 'Commit' and 'Edit Task List', and 'Cancel.' The user will normally choose a task from the list, enter an optional comment, and click the Commit button. This causes the CVW entity to be committed to the repository with the Task and the optional comments as metadata. If the user needs to edit the task list, they can do so with the Edit Task List button, which brings up the Task List editor. And of course if the user clicks 'Cancel,' the checkin is aborted.

Directly beneath the 'Task Completion…' item is a submenu of additional CVW options. For now, these are the same as the TortoiseSVN options.

*Conflicts and Merges*

Sometimes, when attempting to commit an SVN entity, SVN will discover that the starting place for changes to the local SVN entity is older than the current SVN entity in the repository. This situation is called a *conflict*, and it generally means that it is necessary to decide which of each set of changes, if any, should be accepted, and whether it is necessary to make additional changes in light of the fact that the conflict is being resolved. The resulting CVW entity is said to have *merged* changes from the repository and from the work the author has just performed. Figure 33 shows a conflict situation, and Figure 34 (both from Collins-Sussman, Sussman, and Pilato, 2004) shows its resolution.

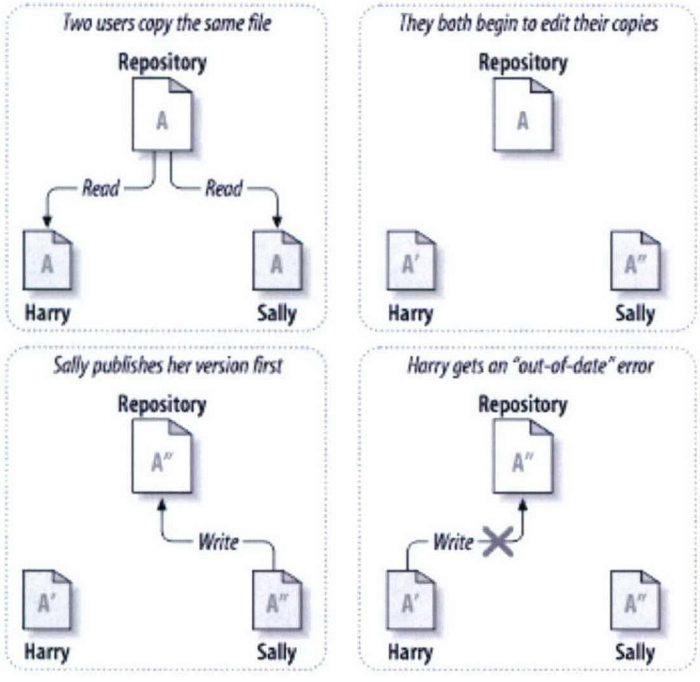



Figure 33. A conflict situation.

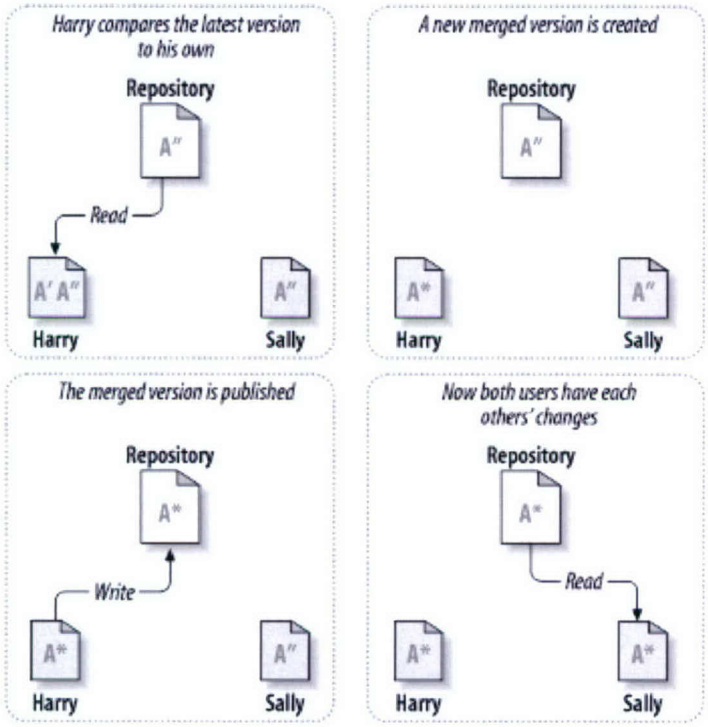



Figure 34. Resolving the conflict.

In the prototype, conflicts are detected at the document (file) level, as provided by the SVN server, and are resolved as follows. The commit transaction that discovered the conflict is aborted, and the conflicting CVW entities are automatically checked out of the repository into the same directory with a slightly different name than the local file.

What happens next depends on the document type. For the prototype, we deal with files of type .doc (that is, MS Word files) only. One local file will automatically be loaded into MS Word, and Word automatically invokes 'Compare and Merge Documents" on the other. The user then manually resolves the conflict by stepping through the differences, making modifications, deletions, and enhancements as necessary, and then saves the file. The saved file is then automatically checked in to the repository with the comment 'Merge of <version> and <version>."

In future releases of the CVW client, other file types will be accommodated.

### Branching and Tagging

Future releases of the prototype will support multiple branches off the version trunk via alternate directories, per the standard SVN practice. Merging of branches back into the trunk will be supported by document-level mechanisms for conflict resolution. There will therefore be no need to add any special features or functionality to the prototype in this area.

### Adaptation to Counterterrorism Analysts

Aptima has had preliminary consultations with Brad Bobenmoyer from ISI Associates about a suitable multiauthor document that counterterrorism analysts commonly prepare. Two suggestions

that emerged were a daily intelligence summary and a situation update report. We decided to focus on the latter since versioning is likely to be more useful for it (old intelligence summaries are of less interest.) Adapting the prototype to this report will primarily involve creation and maintenance of new templates for the CVW client. Aptima will be consulting with ISI Associates extensively in the coming months to make sure we get these templates right.

### The Future

The current prototype focuses primarily on authors' local task lists, which is useful for many purposes, but which also presents the opportunity enhancement by intelligent agents, to be developed under the Cougaar framework. Each author will have an dedicated automated agent to notice user and event context, and will communicate with a central planning agent dedicated to monitoring progress, noticing when any given task has fallen far behind schedule (or is uncovered entirely), and making suggestions to analysts about changes they could make to their local task lists that will improve the timeliness and effectiveness of the mission. In addition, we plan to integrate CVW functionality with Vignette Collaboration Server. If VCS provides enough versioning functionality, that functionality will be used as an underlying version server; otherwise, a third-party version server such as SVN will be used.

## *Task 5. Performance Measurement*

Measurement is an essential component of Topsail. During the past year, the NEMESIS project has been working closely with the metrics contractor, SAIC. We have iterated with them on detailed metrics development and have engaged in an on-going dialogue about measurement for NEMESIS more broadly. Aptima has extensive experience in measuring collaboration and team performance, and we bring this background to the project as well.

In the balance of this section, we address both compatibility with the SAIC Topsail metrics approach and potential implications of Aptima's overall approach to team performance measurement on broader measurement issues. The latter provides what we believe may be useful background and context, beyond the specifics of the SAIC metrics framework. We are fully cooperating with SAIC. During the next year of NEMESIS, we plan to continue working closely with SAIC on metrics, and we anticipate that the extensive list of quantifiers and measurements that have been developed for NEMESIS will continue to evolve. We also intend to contribute to the overall measurement approach.

In our work over the past year, we have focused on "higher-level" metrics, specifically conducting TIES with other Topsail contractors and moving along the migration path into RDEC.

We conducted a successful TIE with Cycorp, described above. Briefly, ORA was able to run on the TKB extract and develop interpretable network representations from it. This TIE shows the potential general utility of ODL as a *lingua franca* between data (or knowledge) sources and analytical tools, although it was only applied to one specific source of knowledge and one tool in this instance.

We have another TIE underway to use a TKB extract as input into ASAM, again mediated (or translated) through ODL. The input requirements for ASAM differ from those for ORA; in particular, ASAM wants "events" as inputs. If this TIE is successful, it will set the stage for further expansion of ODL's scope. In particular, we will explore TIES using either a different data source

than TKB, an end application beyond NEMESIS (of which both this application of ORA and ASAM are components), or both.

We have also begun discussions about TIES using our collaborative workflow infrastructure with Topsail components developed by other contractors. We anticipate initiating at least one such TIE during Q2 2005.

Technologies from NEMESIS are transitioning into the RDEC platform. Both ORA and ASAM have been evaluated in the RDEC Prototype Platform, and ORA will soon be moving to the Data Protected Platform. We plan to submit the CVW prototype to the Prototype Platform in 2005, as well.

Aptima's general approach to the problem of measuring and understanding team performance typically involves three methods: controlled human-in-the-loop experiments, constructive simulation and modeling, and field assessment. In Aptima's typical controlled experiments, test scenarios matching investigatory hypotheses are systematically developed and run with individuals who share characteristics with the eventual users of the system. Quantitative outcomes are then statistically analyzed, and inferences are drawn. When the mission or task is suitably structured, Aptima does rigorous modeling of the team, task structure, and available resources and, in some cases, produces one or more team designs customized for the mission. Finally, results from the laboratory and from modeling are assessed in a live setting to test the hypotheses and models developed in the other phases. The results from any of the three of these approaches can then be followed up with one of the other approaches. Aptima has adopted and applied this approach in a variety of settings, ranging from Command and Control (Entin et al., 2002), to collaborative tools (Carolan et al., 2003) to team training feedback (Alliger et al., 2003).

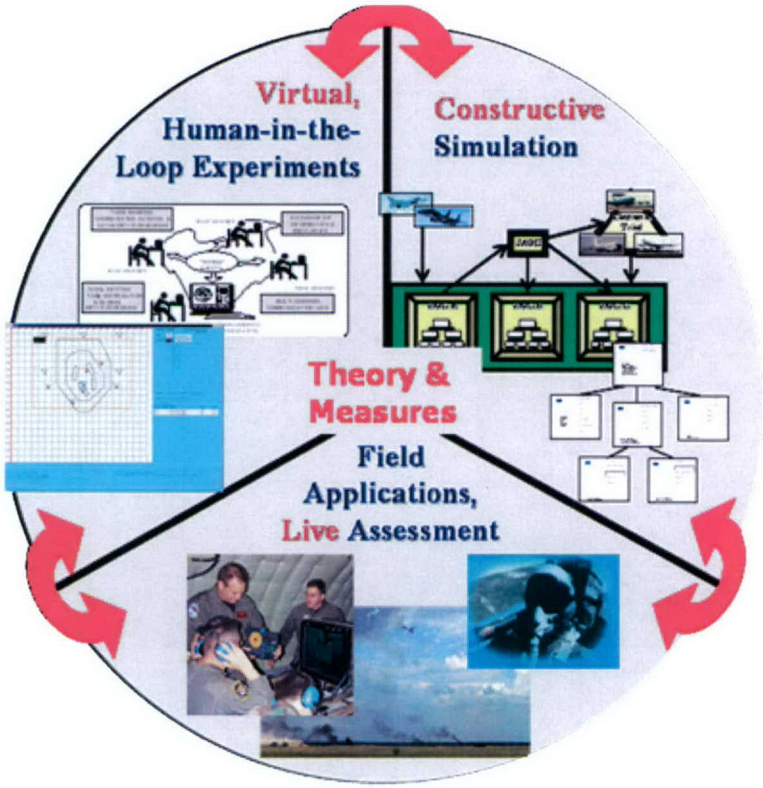



Figure 35. Aptima's approach to team measurement.

The three approaches combine to provide answers to the three major questions one should ask about measurement found in Figure 36: *what* to measure, *when* to measure, and *how* to measure.

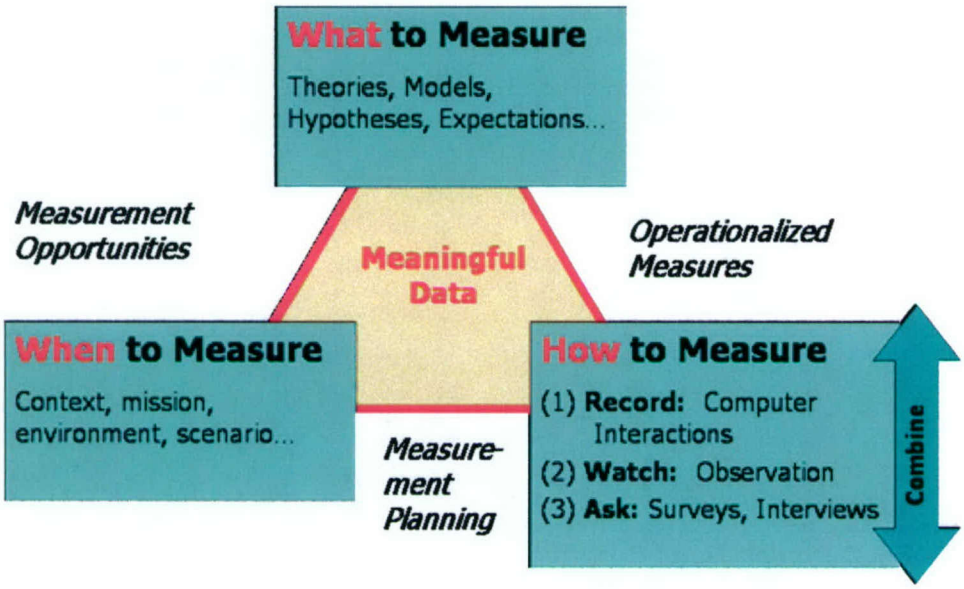



Figure 36. The three pillars of team performance measurement.

We have had fruitful interactions with SAIC about two linked broader measurement issues that flow out of our approach. Both of them concern the bottom-up focus of the Topsail measurement approach thus far. In general, the Topsail measurement approach seeks to automate measurement as much as possible. Lower-level measures will then be rolled up or aggregated (perhaps using complex Boolean functions) into higher-order metrics.

If the only source of information used in assessing performance on the higher-order measures comes from the lower-order measures, then no information not implicit in them can feed into those higher-order results. For example, conceptually, the "effectiveness" of a tool must in some way reflect how well a user of that tool performs the task(s) it supports. (One could even argue that real "effectiveness" requires that the tasks being supported be, in some sense, the "right" tasks – tasks that support the agency and the IC mission. However, this level of consideration is probably outside of the scope of the metrics effort.) But assessing overall effectiveness in performing a task may be hard, or, in some cases, impossible, to extract purely from aggregating up from automated measures.

Although Topsail measurement is not driven by theories or models of the phenomena or processes being supported, implicit models or theories about the relationships of automatable process measures to outcome measures may be embedded in the measurement framework. For example, metrics like "number of historical references to an entity per unit time" or "number of ways time can be searched" seem to imply that more is better. But it is not necessarily the case that having many ways to search by time will help analysts perform their tasks. Some tasks may be facilitated by having only a few, but the right[3] few, ways to search time available. (Along these lines, it has been

[3] In a way, the definition of "right" here is tautological: a method is "right" iff it helps the analyst perform the task. But this tautology suggests the importance of developing task-level measures as well as the potential utility of analyzing analyst tasks in some depth before designing tools, and adding features to those tools, intended to help them.

observed that some people do not always find the additional features and capabilities embodied in each new release of MSWord to actually help them in performing their tasks.)

A related concern can be thought of colloquially as "teaching to the test" (that is, contractors designing their tools to get good scores). For example, if one were being evaluated on "number of ways time can be searched," one might be tempted to develop additional ways to search by time, even without a compelling analysis that tied all such methods to enhanced task performance. We need to be careful that pushing vigorously toward measurement does not inadvertently drive design and development efforts in particular directions. The tendency to try to accomplish what is being measured has been documented across many areas of human endeavor, from elementary school to the top echelons of Fortune 100 companies. Indeed, it is one of the drivers of the emphasis being placed on measurement. But it also suggests taking care that measurement does not have unintended consequences.

## Summary

The second year of the NEMESIS project has extended the first year's solid foundation for developing a state-of-the-art environment that provides an integrated analytical and predictive view of hostile networks, their likely current and future activities, and optimal strategies for disabling or destroying them.

Collaborative versioning and workflow will provide rich capabilities for managing a revisable collection of documents and will provide sophisticated but unobtrusive coordination of missions and tasks involving multiple analysts NEMESIS' bus-oriented integration platform and accompanying SDK provides a foundation that will scale gracefully as new applications are added, and ODL provides a common language for integration, as illustrated by the TIE with Cycorp. The ASAM tool from the University of Connecticut provides an advanced means to identify matches between the pattern of transactions and a library of patterns of hostile activities, to estimate the likeliest current state of those patterns, to manage associated uncertainties, and to find optimal intervention strategies. The ORA tool from Carnegie Mellon University provides a means to understand the vulnerabilities of the hostile network organizations and, in conjunction with other tools from Carnegie Mellon, also to predict the effects of interventions such as disabling or capturing given members in the network.

Subsequent iterations of NEMESIS will refine these capabilities with special attention to the specific requirements of the intelligence community and their collaboration and integration environments. To ensure that the product of this refinement is useful to the intended users, an informative and rigorous measurement program for NEMESIS will continue to be pursued in quasi-laboratory settings as well as quasi-field settings.

We believe the result will be a quantum leap in collaborative, automated capability for analysts. NEMESIS is poised to enable them to leverage each others' knowledge and skills more efficiently with more effective center-edge collaboration, to work faster by automating the management of collaborative artifacts and what-if scenarios, and to work smarter by amplifying the analysts' cognitive processing of patterns of hostile activity and of hostile meta-networks, and to deliver the benefits that can only be obtained from multiple, interactive views of the same situation

## References

Alliger, G., Garrity, M.J., McCall, Beer, L., and Rodriguez, D. (2003). Competency-based Definition of Work and Performance for Command and Control. Paper presented at the *13th International Occupational Analyst Workshop,* San Antonio, TX, April 2003.

Altova (2005). *XMLSpy User Manual and Programmer's Reference*. Altova, Inc.

Boyd D (2002). Faceted Id/entity: Managing representation in a digital world, Master's degree thesis, MIT Media Lab.

Carley, K.M. (2002). *"Smart Agents and Organizations of the Future" The Handbook of New Media*. Edited by Leah Lievrouw and Sonia Livingstone, Ch. 12, pp. 206-220, Thousand Oaks, CA, Sage.

Carley, K.M. (2003) Dynamic Network Analysis. In R. Breiger, Carley, K. and Pattison, P. (Eds.) *Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers*, Washington: The National Academies Press.

Carley, K.M. (2004). "Estimating Vulnerabilities in Large Covert Networks Using Multi-Level Data." *In Proceedings of the 2004 International Symposium on Command and Control Research and Technology*. Conference held in June, San Diego, CA., Evidence Based Research, Presented during Track 1, Electronic Publication, Vienna, VA.

Carley, K.M. and Reminga, J. (2004). ORA: *Organization Risk Analyzer*. Carnegie Mellon University, School of Computer Science, Institute for Software Research International, Technical Report CMU-ISRI-04-101.

Carley, K. M., Fridsma, D., Casman, E., Altman, N., Chang, J., Kaminsky, B., Nave, D., and Yahja, A. (2003). BioWar: Scalable Multi-Agent Social and Epidemiological Simulation of Bioterrorism Events. *Proceedings of the 2003 NAACSOS Conference*. http://www.casos.cs.cmu.edu/publications/working_papers/carley_2003_biowar.pdf.

Carley, K.M. and Hill, V. (2001) "Structural Change and Learning Within Organizations". In Dynamics of Organizations: *Computational Modeling and Organizational Theories*. Edited by Alessandro Lomi and Erik R. Larsen, MIT Press/AAAI Press/Live Oak, Ch. 2. pp 63-92.

Carolan, T., MacMillan, J., Entin, E.B., Morley, R.M., Schreiber, B.T., Portrey, A., Denning, T., and Bennett, W. Jr. (2003). Integrated Performance Measurement and Assessment in Distributed Mission Operations Environments: Relating Measures to Competencies. In *Proceedings of the 25th Interservice/Industry Training, Simulation & Education Conference*, Orlando, FL, December.

Collins-Sussman, B., Fitzpatrick, B.W., and Pilato, C.M. (2004) Version *Control with Subversion.* O'Reilly Media, http://www.oreilly.com.

Cozman, F.G. (1998). The *Interchange Format for Bayesian Networks.* http://www-2.cs.cmu.edu/afs/cs/user/fgcozman/www/Research/InterchangeFormat/.

Entin, E.E., Diedrich, F.J., MacMillan, J. and Serfaty, D. (2002). Awareness and C2 Organizational Structure. In *Proceedings of the Command and Control Research and Technology Symposium*, Monterrey, CA, June 2002.

Harrison, I. (2002). *Pattern Schema Design.* www.ai.sri.com/~law/schemas/2002/07/patternML.doc

Indian Embassy (2002), Information of Indian Hijacked Flight IC-814, http://www.indianembassy.org/archive/IC_814.html.

Peters K. (2004). BIT -101, www.bit-101.com

Potter J. (2003). Buddygraph Project, www.buddygraph.com

Stacy, E.W. (2004). *An Overview of ODL.* Aptima Technical Report AP-R-1154, Aptima, Inc.

Swami, P. (2000). Kashmir after Kandahar, *Frontline Magazine*, *17*(2), Jan.22-Feb.04, 2000, http://www.flonnet.com/fl1702/17020040.htm.

Tsvetovat, M., Reminga, J., & Carley, K.M. (2003). DyNetML: Interchange Format for Rich Social Network Data, *Proceedings of the 2003 NAACSOS Conference.* http://www.casos.cs.cmu.edu/publications/working_papers/tsvetovat_2003_dynetml.pdf.

World Wide Web Consortium (2004). *Modularization of XHTML™ 1.0 – Second Edition.* W3C Working Draft, 18 February 2004. http://www.w3.org/TR/2004/WD-xhtml-modularization-20040218/.

# Appendix A: Publications and Presentations

## Book Chapter:

Krishna Pattipati, Peter Willett, Jeffrey Allanach, Haiying Tu, Satnam Singh, "Hidden Markov Models and Bayesian Networks for Counter-Terrorism", in "21st Century Enabling Technologies and Policies for Counter-Terrorism", Robert Popp and John Yen eds., to be published by IEEE press.

## Journal Papers:

Haiying Tu, Jeffrey Allanach, Satnam Singh, Peter Willett and Krishna Pattipati, "Information Integration via Hierarchical and Hybrid Bayesian Networks", submitted to *IEEE Transactions on System, Man and Cybernetics, Part A: Systems and Humans, special issue on "Advances in Heterogeneous and Complex System Integration"*. December 2004.

Satnam Singh, Haiying Tu, Jeffrey Allanach, Krishna Pattipati and Peter Willett, "Modeling Threats", *IEEE Potentials*, August/September, 2004.

## Conference Papers:

Haiying Tu, Jeffrey Allanach, Satnam Singh, Krishna Pattipati and Peter Willett, "The Adaptive Safety Analysis and Monitoring System", *SPIE Defense and Security Symposium*, April 12-16 2004.

Satnam Singh, Jeffrey Allanach, Haiying Tu, Krishna Pattipati and Peter Willett, "Stochastic Modeling of a Terrorist Event via the ASAM system", *IEEE Conference on Systems, Man and Cybernetics,* The Hague, The Netherlands. Oct 10~13, 2004.

Jeffrey Allanach, Haiying Tu, Satnam Singh, Krishna Pattipati and Peter Willet, "Detecting, Tracking and Counteracting Terrorist Networks via Hidden Markov Models," *IEEE Aerospace Conference*, Big Sky, MT. March 6-13, 2004.

Robert Popp, Krishna Pattipati, Peter Willett, Daniel Serfaty, Webb Stacy, Kathleen Carley, Jeffrey Allanach, Haiying Tu and Satnam Singh, "Collaboration and Modeling Tools for Counter-Terrorism Analysis", *CIHSPS2004 - IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety*, Venice, Italy, 21-22 July 2004

Robert Popp, Krishna Pattipati, Peter Willett, Daniel Serfaty, Webb Stacy, Kathleen Carley, Jeffrey Allanach, Haiying Tu and Satnam Singh, "Collaborative Tools for Counter-Terrorism Analysis", 2005 IEEE Aerospace Conference, Big Sky, MT. March 5-12, 2005.

## Poster:

Satnam Singh, Jeffrey Allanach, Haiying Tu, Krishna Pattipati and Peter Willett, "Stochastic Models for Counter-Terrorism Analysis," *ICATHS 2004*, Univ. of Connecticut, CT, August 2004.

# Appendix B: ODL Documentation

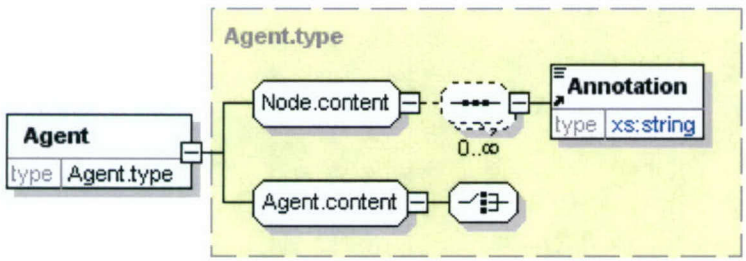
# Schema **odl.xsd**

schema location: C:\Documents and Settings\wstacy\Desktop\Folders\odl\svn\branches\0.6\ODL\Schemas\odl.xsd
targetNamespace: **http://cons.aptima.com/schemas/odl**

| Elements | Groups | Complex types | Simple types | Attr. groups |
|---|---|---|---|---|
| Agent | Agent.content | Agent.type | Date | Agent.attlist |
| Agents | Binding.content | Agents.type | DeltaAction | Binding.attlist |
| Annotation | Communication.content | Binding.type | Mobility | Communication.attlist |
| Binding | Delta.content | Communication.type | NodeTypeEnum | Delta.attlist |
| Communication | Edge.content | Communications.type | PersistentDate | Edge.attlist |
| Communications | Event.content | Delta.type | Probability | Event.attlist |
| Delta | Knowledge.content | Edge.type | | Knowledge.attlist |
| Edge | Location.content | Event.type | | Location.attlist |
| Event | Model.content | Events.type | | Model.attlist |
| Events | Network.content | Knowledge.type | | Network.attlist |
| Knowledge | Node.content | KnowledgeItems.type | | Node.attlist |
| KnowledgeItems | NodeRef.content | Location.type | | NodeRef.attlist |
| Location | Resource.content | Locations.type | | Organization.attlist |
| Locations | Task.content | Model.type | | Resource.attlist |
| Model | | Network.type | | Task.attlist |
| Network | | Networks.type | | |
| Networks | | Node.type | | |
| Node | | NodeRef.type | | |
| NodeRef | | Organization.type | | |
| Organization | | Organizations.type | | |
| Organizations | | Resource.type | | |
| Resource | | Resources.type | | |
| Resources | | Task.type | | |
| Task | | Tasks.type | | |
| Tasks | | | | |

## element **Agent**

diagram




namespace http://cons.aptima.com/schemas/odl
type **Agent.type**
properties content complex
children **Annotation**
used by complexType **Agents.type**

attributes

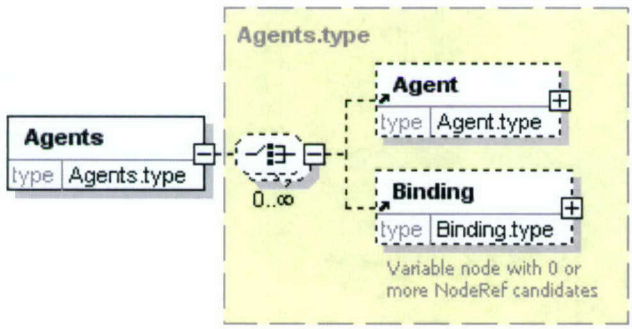
| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | **xs:token** | optional | | | |
| ID | **xs:ID** | required | | | |
| NodeProbability | **Probability** | optional | | | |
| BeginApplicability | **Date** | optional | ALWAYS | | |
| EndApplicability | **Date** | optional | NEVER | | |

source

```
<xs:element name="Agent" type="Agent.type"/>
```

## element **Agents**

diagram




namespace http://cons.aptima.com/schemas/odl

type **Agents.type**

properties content complex

children **Agent** **Binding**

used by group **Model.content**

source

```
<xs:element name="Agents" type="Agents.type"/>
```

## element **Annotation**

diagram


namespace http://cons.aptima.com/schemas/odl

type extension of **xs:string**

properties content complex

used by groups **Edge.content** **Node.content**

attributes

| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Author | **xs:string** | optional | | | |
| Source | **xs:string** | optional | | | |
| DateOfAnalysis | **xs:string** | optional | | | |

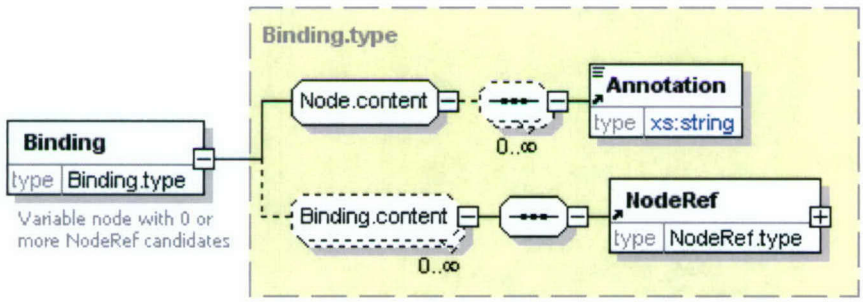
source

```
<xs:element name="Annotation">
 <xs:complexType>
  <xs:simpleContent>
   <xs:extension base="xs:string">
    <xs:attribute name="Author" type="xs:string" use="optional"/>
```

```
      <xs:attribute name="Source" type="xs:string" use="optional"/>
      <xs:attribute name="DateOfAnalysis" type="xs:string" use="optional"/>
     </xs:extension>
    </xs:simpleContent>
   </xs:complexType>
  </xs:element>
```

element **Binding**

diagram




namespace http://cons.aptima.com/schemas/odl

type Binding.type

properties content complex

children Annotation NodeRef

used by complexTypes Agents.type Communications.type Events.type KnowledgeItems.type Locations.type Networks.type Organizations.type Resources.type Tasks.type

attributes

| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | xs:token | optional | | | |
| ID | xs:ID | required | | | |
| NodeProbability | Probability | optional | | | |
| BeginApplicability | Date | optional | ALWAYS | | |
| EndApplicability | Date | optional | NEVER | | |
| VariableClass | NodeTypeEnum | optional | | | |

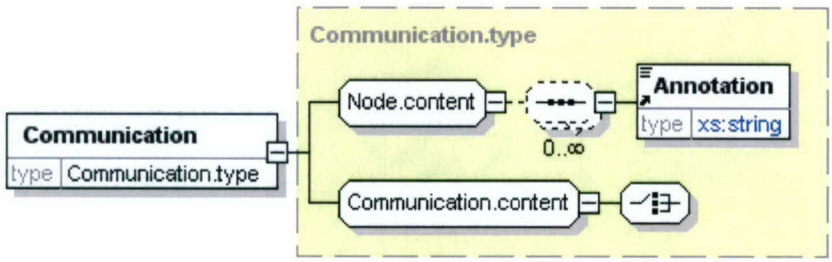
annotation documentation Variable node with 0 or more NodeRef candidates

source

```
<xs:element name="Binding" type="Binding.type">
 <xs:annotation>
  <xs:documentation>Variable node with 0 or more NodeRef
candidates</xs:documentation>
 </xs:annotation>
</xs:element>
```

## element **Communication**

diagram




namespace http://cons.aptima.com/schemas/odl

type **Communication.type**

properties content complex

children **Annotation**

used by complexType **Communications.type**

attributes

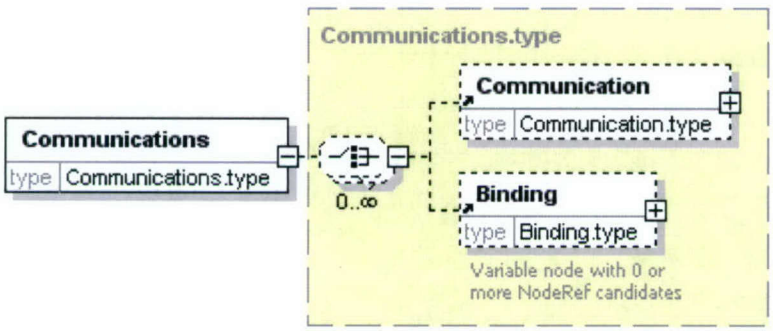
| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | xs:token | optional | | | |
| ID | xs:ID | required | | | |
| NodeProbability | Probability | optional | | | |
| BeginApplicability | Date | optional | ALWAYS | | |
| EndApplicability | Date | optional | NEVER | | |
| From | xs:IDREF | optional | | | |
| To | xs:IDREF | optional | | | |

source

```
<xs:element name="Communication" type="Communication.type"/>
```

## element **Communications**

diagram




namespace http://cons.aptima.com/schemas/odl

type **Communications.type**

properties content complex

children **Communication** **Binding**

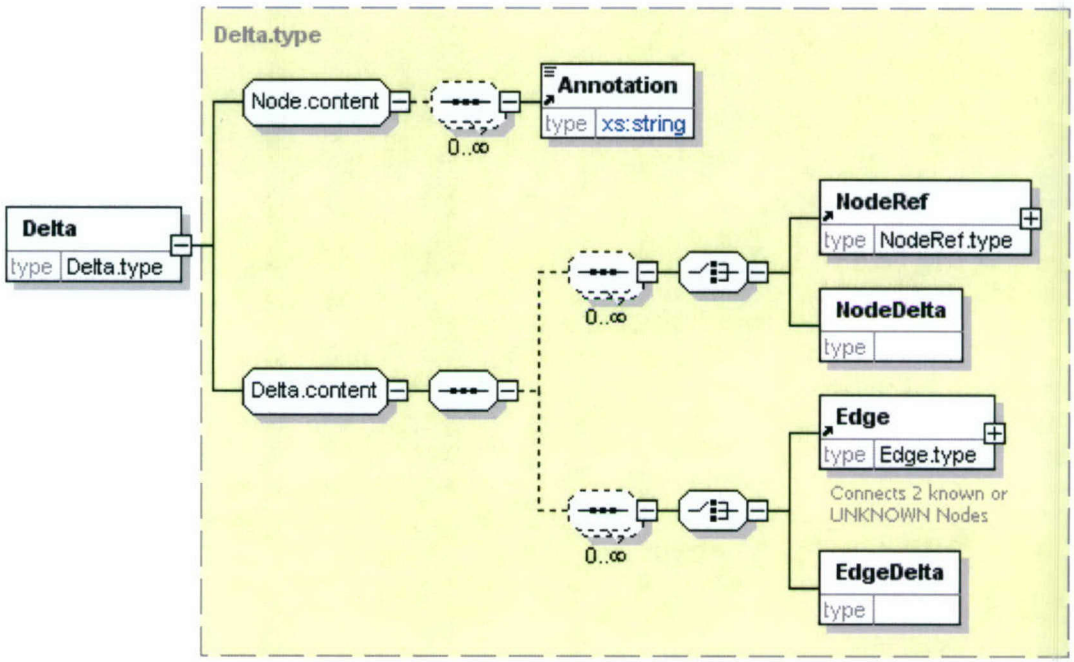
used by group **Model.content**

source

```
<xs:element name="Communications" type="Communications.type"/>
```

## element **Delta**

diagram




namespace http://cons.aptima.com/schemas/odl

type **Delta.type**

properties content complex

children **Annotation** **NodeRef** **NodeDelta** **Edge** **EdgeDelta**

used by complexType **Networks.type**

attributes

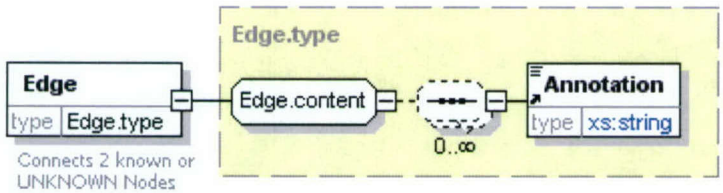
| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | **xs:token** | optional | | | |
| ID | **xs:ID** | required | | | |
| NodeProbability | **Probability** | optional | | | |
| BeginApplicability | **Date** | optional | ALWAYS | | |
| EndApplicability | **Date** | optional | NEVER | | |
| AppliesTo | **xs:IDREF** | required | | | |

source

```
<xs:element name="Delta" type="Delta.type"/>
```

## element **Edge**

diagram




namespace http://cons.aptima.com/schemas/odl

type **Edge.type**

properties content complex

children **Annotation**

used by element **Network.content/Edgelist**
group **Delta.content**

attributes

| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| EdgeProbability | **Probability** | optional | | | |
| TransitionProbability | **Probability** | optional | | | |
| BeginApplicability | **Date** | optional | ALWAYS | | |
| EndApplicability | **Date** | optional | NEVER | | |
| Source | **xs:IDREF** | required | | | |
| Dest | **xs:IDREF** | required | | | |
| Name | **xs:string** | optional | | | |
| Color | **xs:NMTOKEN** | optional | NONE | | |
| ID | **xs:ID** | required | | | |

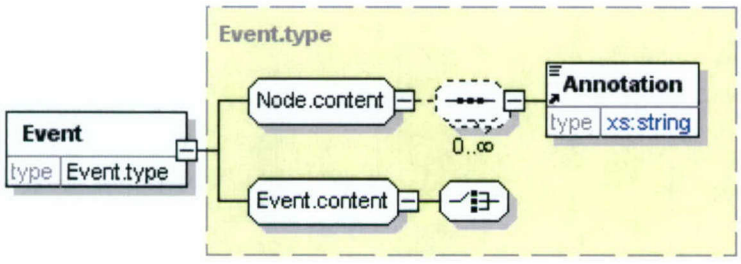
annotation documentation Connects 2 known or UNKNOWN Nodes

source

```
<xs:element name="Edge" type="Edge.type">
 <xs:annotation>
  <xs:documentation>Connects 2 known or UNKNOWN
Nodes</xs:documentation>
 </xs:annotation>
</xs:element>
```

## element **Event**

diagram




namespace http://cons.aptima.com/schemas/odl

type **Event.type**

properties content complex

children **Annotation**

used by complexType **Events.type**

attributes

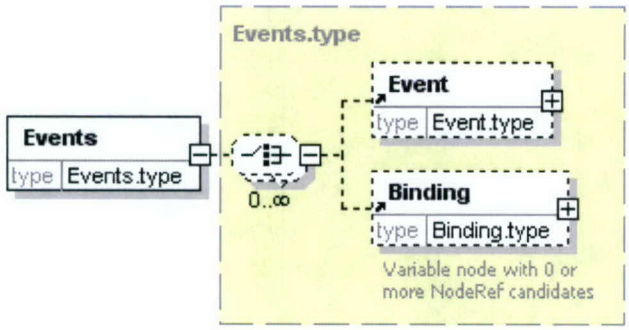
| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | **xs:token** | optional | | | |
| ID | **xs:ID** | required | | | |
| NodeProbability | **Probability** | optional | | | |
| BeginApplicability | **Date** | optional | ALWAYS | | |
| EndApplicability | **Date** | optional | NEVER | | |
| Goal | **xs:IDREF** | optional | | | |

source

```
<xs:element name="Event" type="Event.type"/>
```

## element **Events**

diagram




namespace http://cons.aptima.com/schemas/odl

type **Events.type**

properties content complex

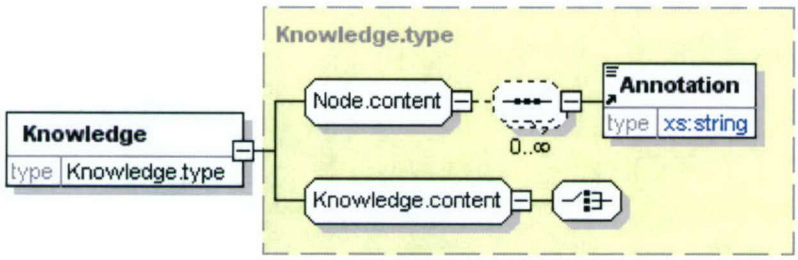
children **Event** **Binding**

used by group **Model.content**

source
```
<xs:element name="Events" type="Events.type"/>
```

## element **Knowledge**

diagram




namespace http://cons.aptima.com/schemas/odl

type **Knowledge.type**

properties content complex

children **Annotation**

used by complexType **KnowledgeItems.type**

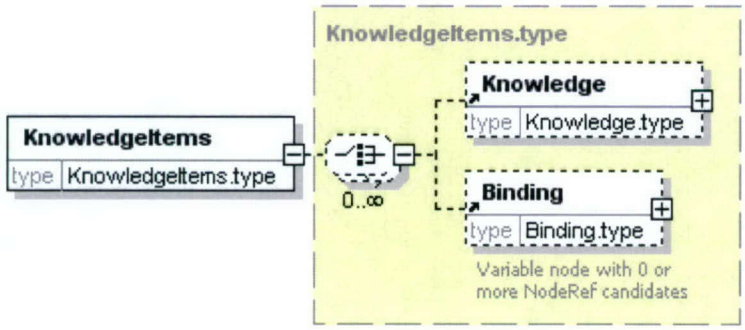
attributes

| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | **xs:token** | optional | | | |
| ID | **xs:ID** | required | | | |
| NodeProbability | **Probability** | optional | | | |
| BeginApplicability | **Date** | optional | ALWAYS | | |
| EndApplicability | **Date** | optional | NEVER | | |
| Fact | **xs:string** | optional | | | |
| KnowledgeBase | **xs:IDREFS** | optional | | | |

source
```
<xs:element name="Knowledge" type="Knowledge.type"/>
```

## element **KnowledgeItems**

diagram




namespace http://cons.aptima.com/schemas/odl

type **KnowledgeItems.type**

properties content complex

children **Knowledge** **Binding**

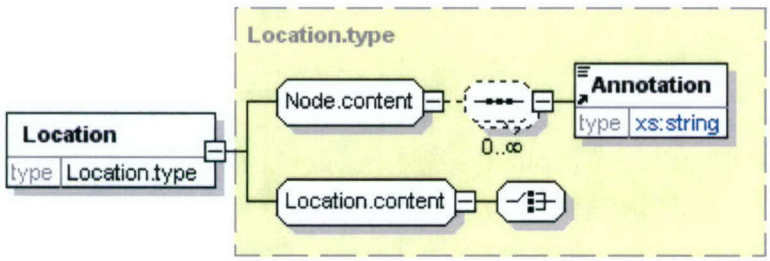
used by group **Model.content**

source

```
<xs:element name="KnowledgeItems" type="KnowledgeItems.type"/>
```

## element **Location**

diagram




namespace http://cons.aptima.com/schemas/odl

type **Location.type**

properties content complex

children **Annotation**

used by complexType **Locations.type**

attributes

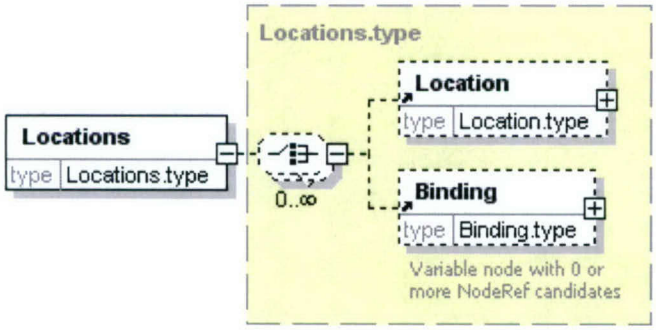
| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | **xs:token** | optional | | | |
| ID | **xs:ID** | required | | | |
| NodeProbability | **Probability** | optional | | | |
| BeginApplicability | **Date** | optional | ALWAYS | | |
| EndApplicability | **Date** | optional | NEVER | | |

source

```
<xs:element name="Location" type="Location.type"/>
```

element **Locations**

diagram




namespace http://cons.aptima.com/schemas/odl

type **Locations.type**

properties content complex

children **Location** **Binding**

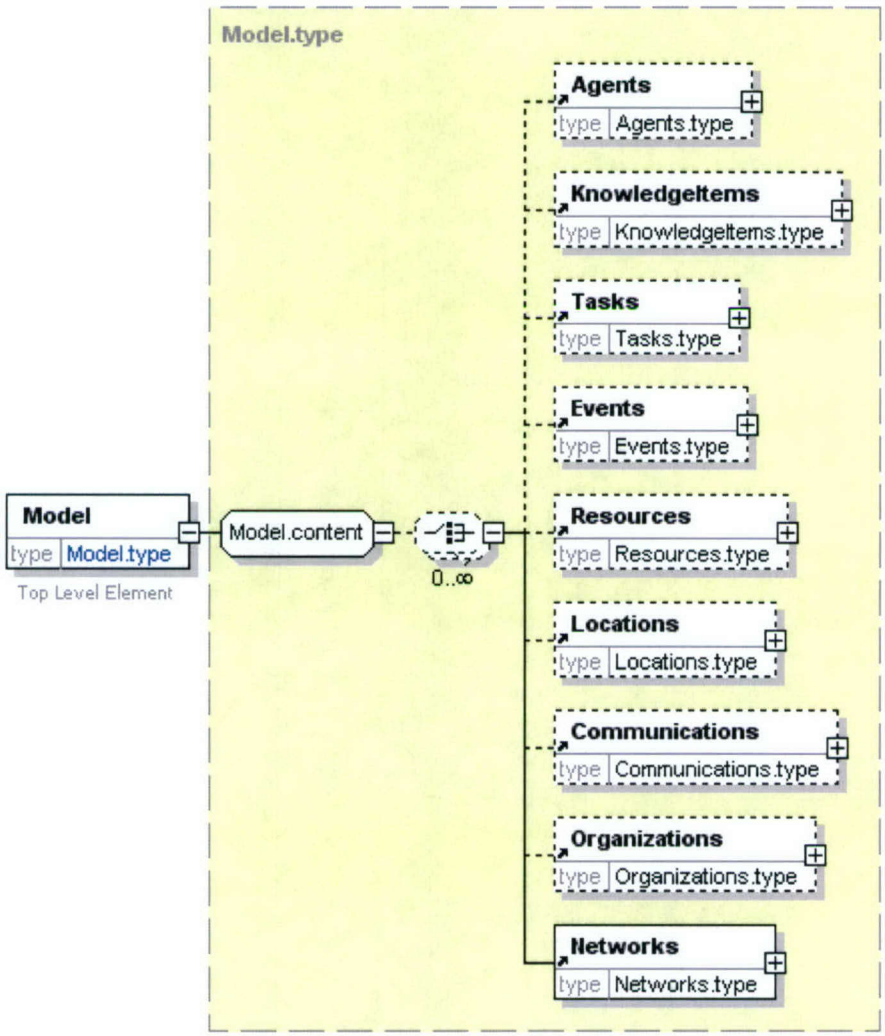
used by group **Model.content**

source

```
<xs:element name="Locations" type="Locations.type"/>
```

element **Model**

diagram



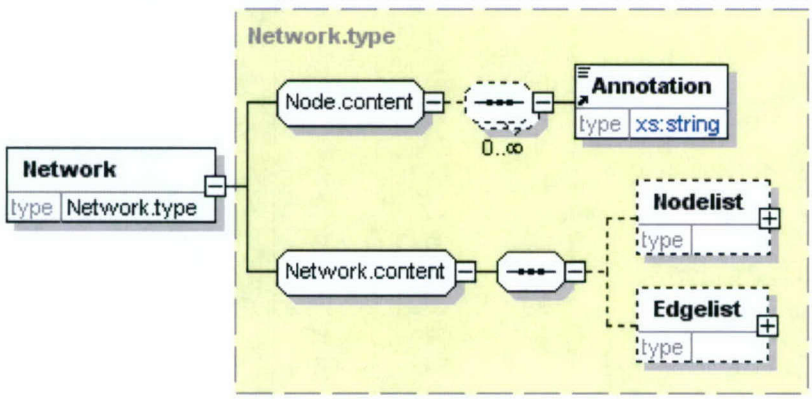

| | | | | | |
|---|---|---|---|---|---|
| namespace | http://cons.aptima.com/schemas/odl | | | | |
| type | extension of **Model.type** | | | | |
| properties | content complex | | | | |
| children | **Agents KnowledgeItems Tasks Events Resources Locations Communications Organizations Networks** | | | | |
| identity constraints | | Name | Refer | Selector | Field(s) |
| | key | Edge | | Model/Organization/Edge | @Source @Dest @Color @BeginApplicability @EndApplicability |
| | keyref | Edge-Ref | Edge | Model/Delta/EdgeDelta | @Source @Dest @Color @BeginApplicability @EndApplicability |
| annotation | documentation Top Level Element | | | | |

source

```
<xs:element name="Model">
 <xs:annotation>
  <xs:documentation>Top Level Element</xs:documentation>
 </xs:annotation>
 <xs:complexType>
  <xs:complexContent>
   <xs:extension base="Model.type"/>
  </xs:complexContent>
 </xs:complexType>
 <xs:key name="Edge">
  <xs:selector xpath="Model/Organization/Edge"/>
  <xs:field xpath="@Source"/>
  <xs:field xpath="@Dest"/>
  <xs:field xpath="@Color"/>
  <xs:field xpath="@BeginApplicability"/>
  <xs:field xpath="@EndApplicability"/>
 </xs:key>
 <xs:keyref name="Edge-Ref" refer="Edge">
  <xs:selector xpath="Model/Delta/EdgeDelta"/>
  <xs:field xpath="@Source"/>
  <xs:field xpath="@Dest"/>
  <xs:field xpath="@Color"/>
  <xs:field xpath="@BeginApplicability"/>
  <xs:field xpath="@EndApplicability"/>
 </xs:keyref>
</xs:element>
```

element **Network**

diagram




namespace http://cons.aptima.com/schemas/odl

type **Network.type**

properties content complex

children **Annotation** **Nodelist** **Edgelist**

used by complexType **Networks.type**

attributes

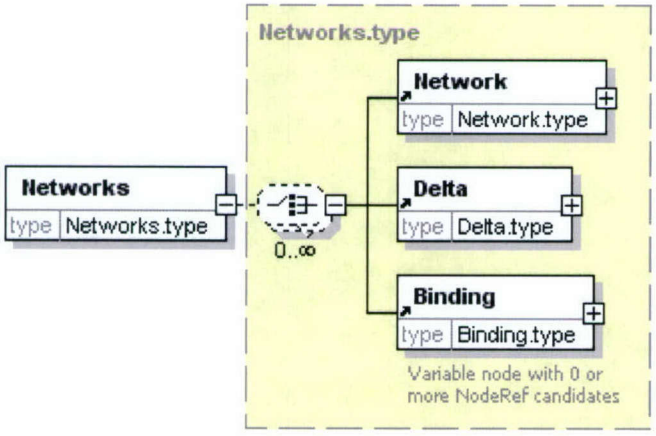
| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | **xs:token** | optional | | | |
| ID | **xs:ID** | required | | | |
| NodeProbability | **Probability** | optional | | | |
| BeginApplicability | **Date** | optional | ALWAYS | | |
| EndApplicability | **Date** | optional | NEVER | | |

source

```
<xs:element name="Network" type="Network.type"/>
```

## element **Networks**

diagram




namespace http://cons.aptima.com/schemas/odl

type **Networks.type**

properties content complex

children **Network** **Delta** **Binding**

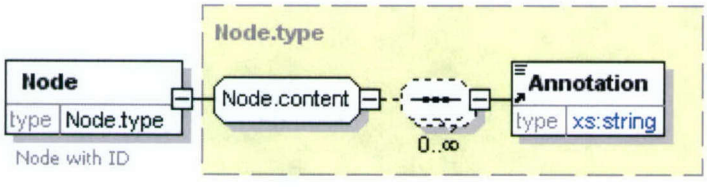
used by group **Model.content**

source

```
<xs:element name="Networks" type="Networks.type"/>
```

## element **Node**

diagram




namespace http://cons.aptima.com/schemas/odl

type **Node.type**

properties content complex

children **Annotation**

attributes

| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | **xs:token** | optional | | | |
| ID | **xs:ID** | required | | | |

| | | | | |
|---|---|---|---|---|
| NodeProbability | Probability | optional | | |
| BeginApplicability | Date | optional | ALWAYS | |
| EndApplicability | Date | optional | NEVER | |

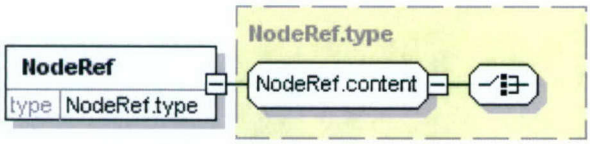
annotation
documentation Node with ID

source

```
<xs:element name="Node" type="Node.type">
 <xs:annotation>
  <xs:documentation>Node with ID</xs:documentation>
 </xs:annotation>
</xs:element>
```

## element **NodeRef**

diagram




namespace http://cons.aptima.com/schemas/odl

type NodeRef.type

properties content complex

used by elements Network.content/Nodelist Resource.content/Platform
groups Binding.content Delta.content

attributes

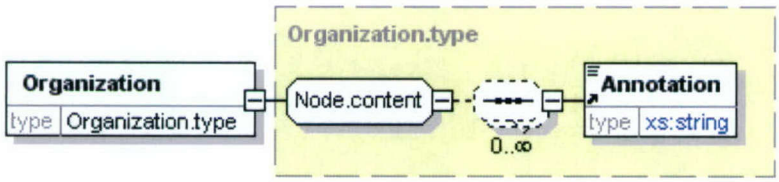
| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| ProbabilityOfNode | Probability | optional | 1 | | |
| Name | xs:string | optional | | | |
| Node | xs:IDREF | required | | | |

source

```
<xs:element name="NodeRef" type="NodeRef.type"/>
```

## element **Organization**

diagram




namespace http://cons.aptima.com/schemas/odl

type Organization.type

properties content complex

children Annotation

used by complexType Organizations.type

attributes

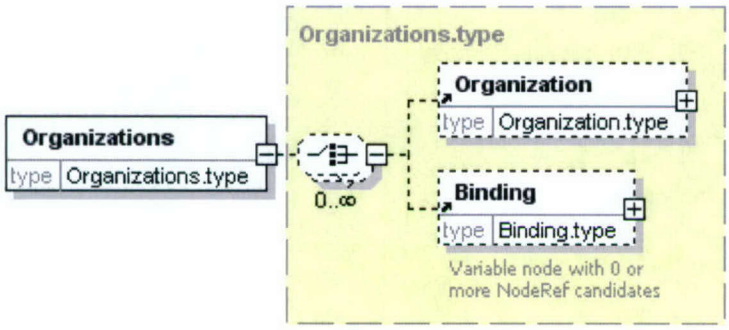
| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | xs:token | optional | | | |
| ID | xs:ID | required | | | |
| NodeProbability | Probability | optional | | | |
| BeginApplicability | Date | optional | ALWAYS | | |
| EndApplicability | Date | optional | NEVER | | |

| | | | |
|---|---|---|---|
| | Network | xs:IDREF | optional |

source
```
<xs:element name="Organization" type="Organization.type"/>
```

## element **Organizations**

diagram




namespace http://cons.aptima.com/schemas/odl

type **Organizations.type**

properties content complex

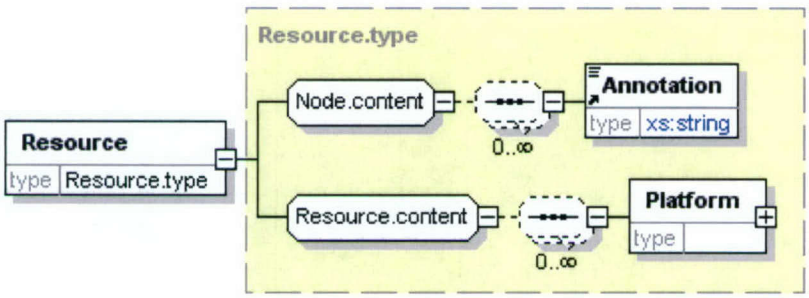
children **Organization** **Binding**

used by group **Model.content**

source
```
<xs:element name="Organizations" type="Organizations.type"/>
```

## element **Resource**

diagram




namespace http://cons.aptima.com/schemas/odl

type **Resource.type**

properties content complex

children **Annotation** **Platform**

used by complexType **Resources.type**

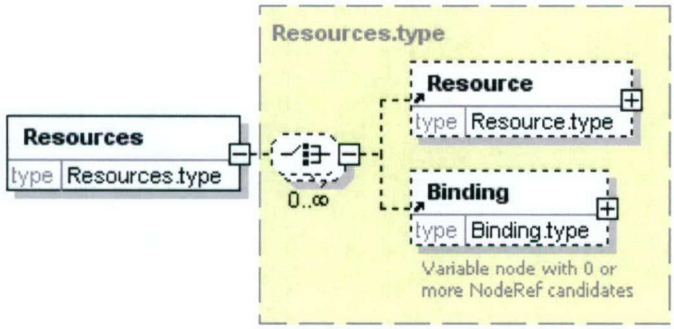
attributes

| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | **xs:token** | optional | | | |
| ID | **xs:ID** | required | | | |
| NodeProbability | **Probability** | optional | | | |
| BeginApplicability | **Date** | optional | ALWAYS | | |
| EndApplicability | **Date** | optional | NEVER | | |

source <xs:element name="Resource" type="Resource.type"/>

element **Resources**

diagram




namespace http://cons.aptima.com/schemas/odl

type **Resources.type**

properties content complex

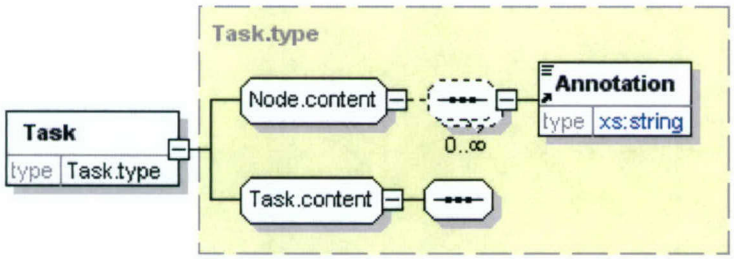
children **Resource** **Binding**

used by group **Model.content**

source <xs:element name="Resources" type="Resources.type"/>

element **Task**

diagram




namespace http://cons.aptima.com/schemas/odl

type **Task.type**

properties content complex

children **Annotation**

used by complexType **Tasks.type**

attributes

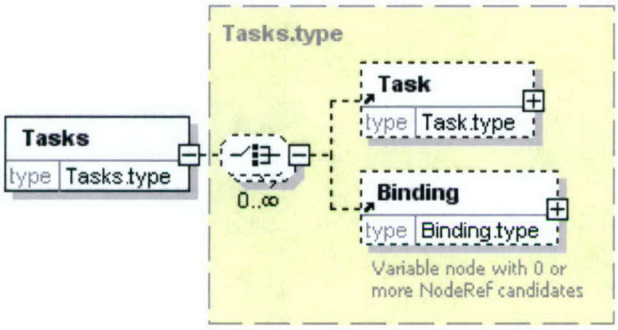
| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | **xs:token** | optional | | | |
| ID | **xs:ID** | required | | | |
| NodeProbability | **Probability** | optional | | | |
| BeginApplicability | **Date** | optional | ALWAYS | | |
| EndApplicability | **Date** | optional | NEVER | | |
| PlannedStart | **xs:dateTime** | optional | | | |
| PlannedEnd | **xs:dateTime** | optional | | | |
| ActualStart | **xs:dateTime** | optional | | | |
| ActualEnd | **xs:dateTime** | optional | | | |

Dependencies xs:IDREFS optional

source
```
<xs:element name="Task" type="Task.type"/>
```

element **Tasks**

diagram




namespace http://cons.aptima.com/schemas/odl

type **Tasks.type**

properties content complex

children **Task** **Binding**

used by group **Model.content**

source
```
<xs:element name="Tasks" type="Tasks.type"/>
```

group **Agent.content**

diagram

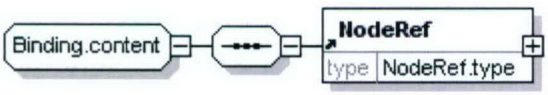

namespace http://cons.aptima.com/schemas/odl

used by complexType **Agent.type**

source
```
<xs:group name="Agent.content">
  <xs:choice/>
</xs:group>
```

group **Binding.content**

diagram




namespace http://cons.aptima.com/schemas/odl

children **NodeRef**

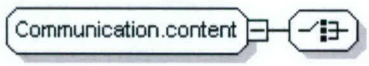
used by complexType **Binding.type**

source
```
<xs:group name="Binding.content">
  <xs:sequence>
```

```
    <xs:element ref="NodeRef"/>
   </xs:sequence>
  </xs:group>
```

group **Communication.content**

diagram




namespace http://cons.aptima.com/schemas/odl

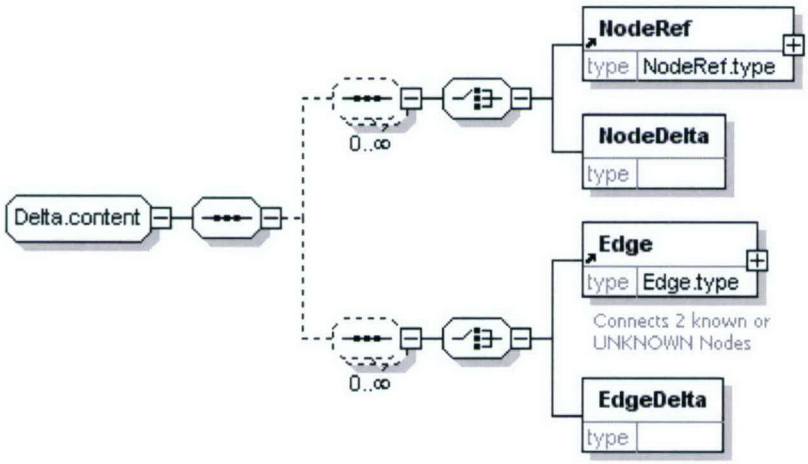
used by complexType **Communication.type**

source

```
<xs:group name="Communication.content">
 <xs:choice/>
</xs:group>
```

group **Delta.content**

diagram




namespace http://cons.aptima.com/schemas/odl

children **NodeRef** **NodeDelta** **Edge** **EdgeDelta**

used by complexType **Delta.type**

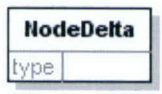
source

```
<xs:group name="Delta.content">
 <xs:sequence>
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
   <xs:choice>
    <xs:element ref="NodeRef"/>
    <xs:element name="NodeDelta">
     <xs:complexType>
      <xs:attribute name="Node" type="xs:IDREF" use="required"/>
      <xs:attribute name="Action" type="DeltaAction" use="optional"
default="DELETE"/>
     </xs:complexType>
```

```
      </xs:element>
     </xs:choice>
    </xs:sequence>
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
     <xs:choice>
      <xs:element ref="Edge"/>
      <xs:element name="EdgeDelta">
       <xs:complexType>
        <xs:attribute name="Edge" type="xs:IDREF" use="required"/>
        <xs:attribute name="Action" type="DeltaAction" use="optional"
default="DELETE"/>
       </xs:complexType>
      </xs:element>
     </xs:choice>
    </xs:sequence>
   </xs:sequence>
</xs:group>
```

element **Delta.content/NodeDelta**

diagram




namespace http://cons.aptima.com/schemas/odl

properties
isRef 0
content complex

attributes

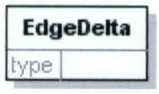
| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Node | xs:IDREF | required | | | |
| Action | DeltaAction | optional | DELETE | | |

source

```
<xs:element name="NodeDelta">
 <xs:complexType>
  <xs:attribute name="Node" type="xs:IDREF" use="required"/>
  <xs:attribute name="Action" type="DeltaAction" use="optional"
default="DELETE"/>
 </xs:complexType>
</xs:element>
```

element **Delta.content/EdgeDelta**

diagram




namespace http://cons.aptima.com/schemas/odl

properties
isRef 0
content complex

attributes

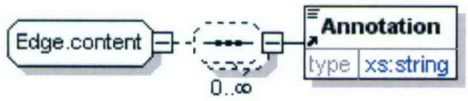
| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Edge | xs:IDREF | required | | | |

| | | | | |
|---|---|---|---|---|
| Action | DeltaAction | optional | DELETE |

source
```
<xs:element name="EdgeDelta">
 <xs:complexType>
  <xs:attribute name="Edge" type="xs:IDREF" use="required"/>
  <xs:attribute name="Action" type="DeltaAction" use="optional"
default="DELETE"/>
 </xs:complexType>
</xs:element>
```

group **Edge.content**

diagram




namespace http://cons.aptima.com/schemas/odl

children Annotation

used by complexType Edge.type

source
```
<xs:group name="Edge.content">
 <xs:sequence minOccurs="0" maxOccurs="unbounded">
  <xs:element ref="Annotation"/>
 </xs:sequence>
</xs:group>
```

group **Event.content**

diagram


namespace http://cons.aptima.com/schemas/odl

used by complexType Event.type

source
```
<xs:group name="Event.content">
 <xs:choice/>
</xs:group>
```

group **Knowledge.content**

diagram


namespace http://cons.aptima.com/schemas/odl

used by complexType Knowledge.type

source
```
<xs:group name="Knowledge.content">
 <xs:choice/>
</xs:group>
```

group **Location.content**

diagram


namespace http://cons.aptima.com/schemas/odl

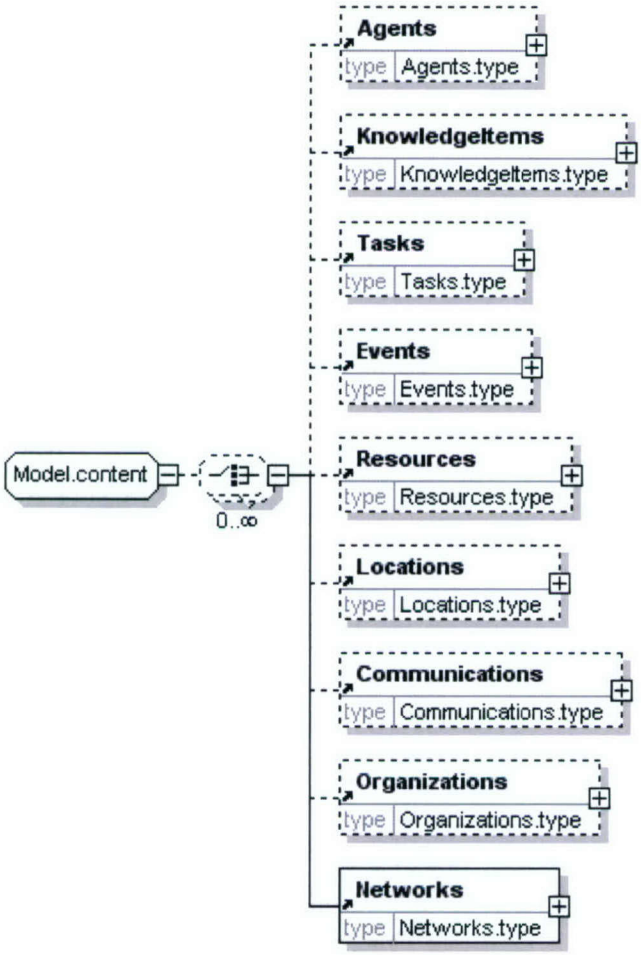
used by complexType **Location.type**

source

```
<xs:group name="Location.content">
  <xs:choice/>
</xs:group>
```

group **Model.content**

diagram




namespace http://cons.aptima.com/schemas/odl

children **Agents** **KnowledgeItems** **Tasks** **Events** **Resources** **Locations** **Communications** **Organizations** **Networks**

used by complexType **Model.type**

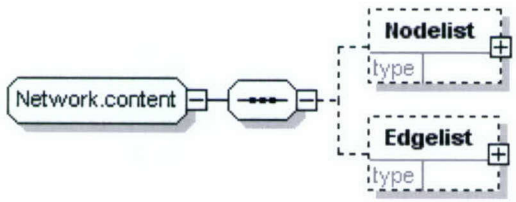
source

```
<xs:group name="Model.content">
```

```
  <xs:choice minOccurs="0" maxOccurs="unbounded">
   <xs:element ref="Agents" minOccurs="0"/>
   <xs:element ref="KnowledgeItems" minOccurs="0"/>
   <xs:element ref="Tasks" minOccurs="0"/>
   <xs:element ref="Events" minOccurs="0"/>
   <xs:element ref="Resources" minOccurs="0"/>
   <xs:element ref="Locations" minOccurs="0"/>
   <xs:element ref="Communications" minOccurs="0"/>
   <xs:element ref="Organizations" minOccurs="0"/>
   <xs:element ref="Networks"/>
  </xs:choice>
 </xs:group>
```

group **Network.content**

diagram




namespace http://cons.aptima.com/schemas/odl

children Nodelist Edgelist

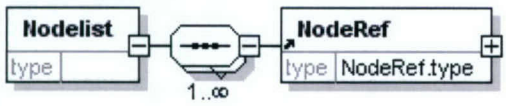
used by complexType Network.type

source

```
<xs:group name="Network.content">
 <xs:sequence>
  <xs:element name="Nodelist" minOccurs="0">
   <xs:complexType>
    <xs:sequence maxOccurs="unbounded">
     <xs:element ref="NodeRef"/>
    </xs:sequence>
   </xs:complexType>
  </xs:element>
  <xs:element name="Edgelist" minOccurs="0">
   <xs:complexType>
    <xs:sequence maxOccurs="unbounded">
     <xs:element ref="Edge"/>
    </xs:sequence>
   </xs:complexType>
  </xs:element>
 </xs:sequence>
</xs:group>
```

element **Network.content/Nodelist**

diagram




namespace http://cons.aptima.com/schemas/odl

properties
isRef 0
content complex

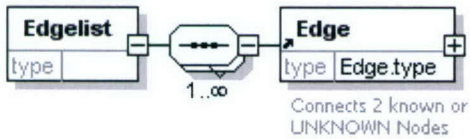
children NodeRef

source

```
<xs:element name="Nodelist" minOccurs="0">
 <xs:complexType>
  <xs:sequence maxOccurs="unbounded">
   <xs:element ref="NodeRef"/>
  </xs:sequence>
 </xs:complexType>
</xs:element>
```

element **Network.content/Edgelist**

diagram




namespace http://cons.aptima.com/schemas/odl

properties
isRef 0
content complex

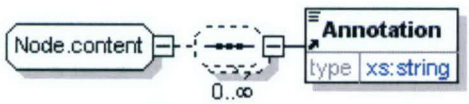
children Edge

source

```
<xs:element name="Edgelist" minOccurs="0">
 <xs:complexType>
  <xs:sequence maxOccurs="unbounded">
   <xs:element ref="Edge"/>
  </xs:sequence>
 </xs:complexType>
</xs:element>
```

group **Node.content**

diagram




namespace http://cons.aptima.com/schemas/odl

children Annotation

used by complexType **Node.type**

source
```
<xs:group name="Node.content">
 <xs:sequence minOccurs="0" maxOccurs="unbounded">
  <xs:element ref="Annotation"/>
 </xs:sequence>
</xs:group>
```

group **NodeRef.content**

diagram

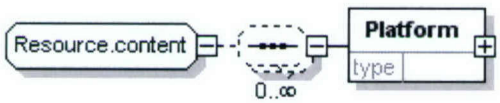

namespace http://cons.aptima.com/schemas/odl

used by complexType **NodeRef.type**

source
```
<xs:group name="NodeRef.content">
 <xs:choice/>
</xs:group>
```

group **Resource.content**

diagram




namespace http://cons.aptima.com/schemas/odl

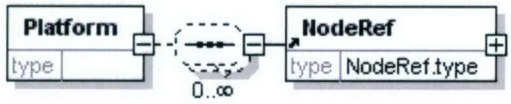
children **Platform**

used by complexType **Resource.type**

source
```
<xs:group name="Resource.content">
 <xs:sequence minOccurs="0" maxOccurs="unbounded">
  <xs:element name="Platform">
   <xs:complexType>
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
     <xs:element ref="NodeRef"/>
    </xs:sequence>
    <xs:attribute name="Name" type="xs:string" use="optional"/>
    <xs:attribute name="Mobility" type="Mobility" use="optional"/>
   </xs:complexType>
  </xs:element>
 </xs:sequence>
</xs:group>
```

element **Resource.content/Platform**

diagram




namespace http://cons.aptima.com/schemas/odl

properties
isRef 0
content complex

children NodeRef

attributes

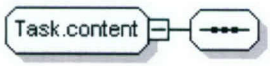
| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | xs:string | optional | | | |
| Mobility | Mobility | optional | | | |

source

```
<xs:element name="Platform">
 <xs:complexType>
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
   <xs:element ref="NodeRef"/>
  </xs:sequence>
  <xs:attribute name="Name" type="xs:string" use="optional"/>
  <xs:attribute name="Mobility" type="Mobility" use="optional"/>
 </xs:complexType>
</xs:element>
```

group **Task.content**

diagram




namespace http://cons.aptima.com/schemas/odl

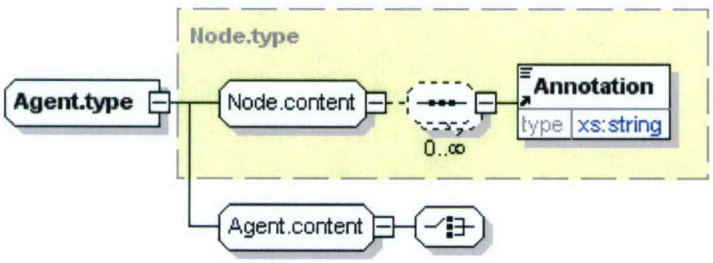
used by complexType Task.type

source

```
<xs:group name="Task.content">
 <xs:sequence/>
</xs:group>
```

complexType **Agent.type**

diagram




namespace http://cons.aptima.com/schemas/odl

type extension of Node.type

properties base Node.type

children Annotation

used by element Agent

attributes

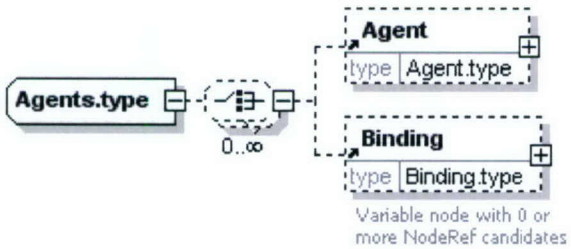
| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | xs:token | optional | | | |
| ID | xs:ID | required | | | |
| NodeProbability | Probability | optional | | | |
| BeginApplicability | Date | optional | ALWAYS | | |
| EndApplicability | Date | optional | NEVER | | |

source

```
<xs:complexType name="Agent.type">
 <xs:complexContent>
  <xs:extension base="Node.type">
   <xs:group ref="Agent.content"/>
   <xs:attributeGroup ref="Agent.attlist"/>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

complexType **Agents.type**

diagram




namespace http://cons.aptima.com/schemas/odl

children Agent Binding

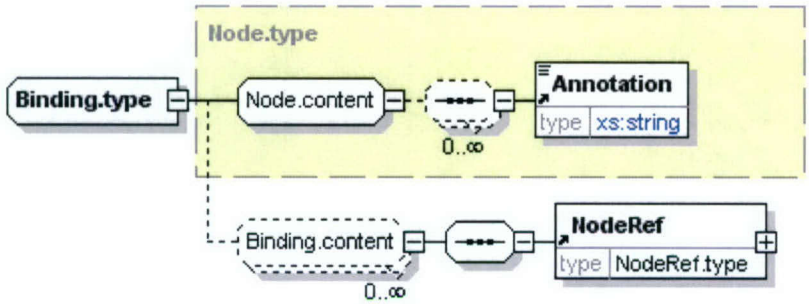
used by element Agents

source

```
<xs:complexType name="Agents.type">
 <xs:choice minOccurs="0" maxOccurs="unbounded">
  <xs:element ref="Agent" minOccurs="0"/>
  <xs:element ref="Binding" minOccurs="0"/>
 </xs:choice>
</xs:complexType>
```

complexType **Binding.type**

diagram




namespace http://cons.aptima.com/schemas/odl

type extension of **Node.type**

properties base Node.type

children **Annotation** **NodeRef**

used by element **Binding**

attributes

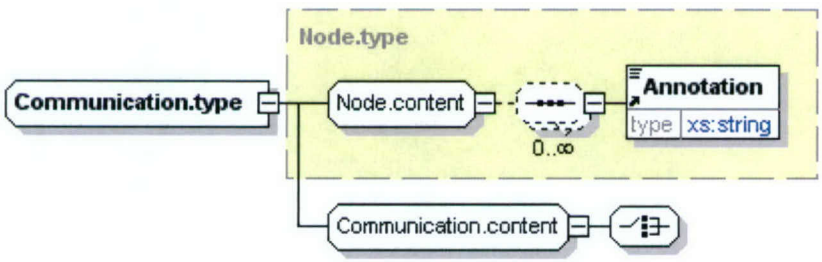
| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | **xs:token** | optional | | | |
| ID | **xs:ID** | required | | | |
| NodeProbability | **Probability** | optional | | | |
| BeginApplicability | **Date** | optional | ALWAYS | | |
| EndApplicability | **Date** | optional | NEVER | | |
| VariableClass | **NodeTypeEnum** | optional | | | |

source

```
<xs:complexType name="Binding.type">
 <xs:complexContent>
  <xs:extension base="Node.type">
   <xs:group ref="Binding.content" minOccurs="0" maxOccurs="unbounded"/>
   <xs:attributeGroup ref="Binding.attlist"/>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

complexType **Communication.type**

diagram




namespace http://cons.aptima.com/schemas/odl

type extension of **Node.type**

properties base Node.type

children Annotation

used by element Communication

attributes

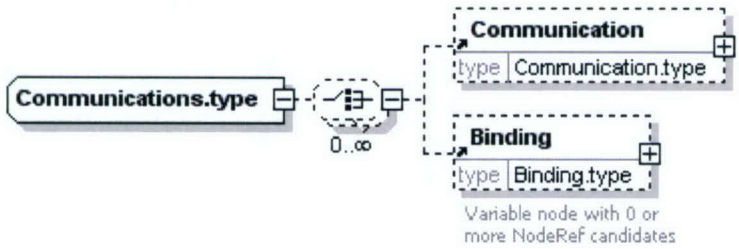
| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | xs:token | optional | | | |
| ID | xs:ID | required | | | |
| NodeProbability | Probability | optional | | | |
| BeginApplicability | Date | optional | ALWAYS | | |
| EndApplicability | Date | optional | NEVER | | |
| From | xs:IDREF | optional | | | |
| To | xs:IDREF | optional | | | |

source

```
<xs:complexType name="Communication.type">
 <xs:complexContent>
  <xs:extension base="Node.type">
   <xs:group ref="Communication.content"/>
   <xs:attributeGroup ref="Communication.attlist"/>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

complexType **Communications.type**

diagram




namespace http://cons.aptima.com/schemas/odl

children Communication Binding

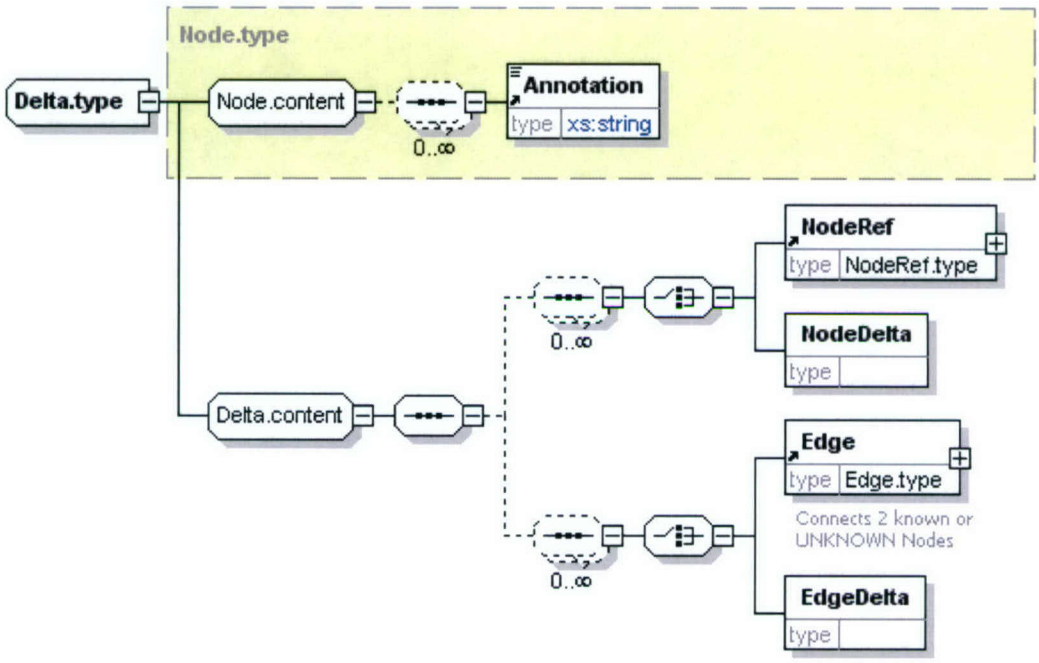
used by element Communications

source

```
<xs:complexType name="Communications.type">
 <xs:choice minOccurs="0" maxOccurs="unbounded">
  <xs:element ref="Communication" minOccurs="0"/>
  <xs:element ref="Binding" minOccurs="0"/>
 </xs:choice>
</xs:complexType>
```

## complexType **Delta.type**

diagram




namespace http://cons.aptima.com/schemas/odl

type extension of **Node.type**

properties base Node.type

children **Annotation** **NodeRef** **NodeDelta** **Edge** **EdgeDelta**

used by element **Delta**

attributes

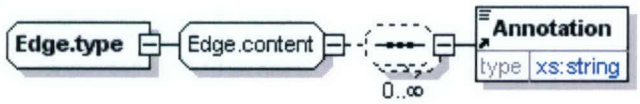
| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | **xs:token** | optional | | | |
| ID | **xs:ID** | required | | | |
| NodeProbability | **Probability** | optional | | | |
| BeginApplicability | **Date** | optional | ALWAYS | | |
| EndApplicability | **Date** | optional | NEVER | | |
| AppliesTo | **xs:IDREF** | required | | | |

source

```
<xs:complexType name="Delta.type">
 <xs:complexContent>
  <xs:extension base="Node.type">
   <xs:group ref="Delta.content"/>
   <xs:attributeGroup ref="Delta.attlist"/>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

## complexType **Edge.type**

diagram




namespace http://cons.aptima.com/schemas/odl

children **Annotation**

used by element **Edge**

attributes

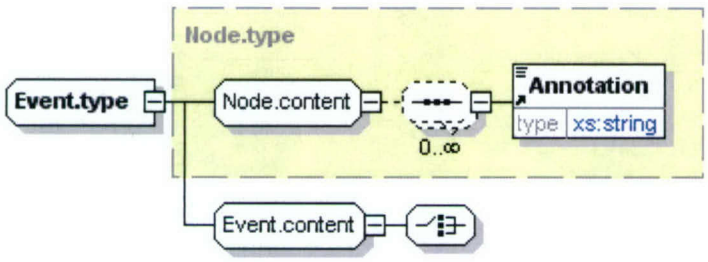
| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| EdgeProbability | **Probability** | optional | | | |
| TransitionProbability | **Probability** | optional | | | |
| BeginApplicability | **Date** | optional | ALWAYS | | |
| EndApplicability | **Date** | optional | NEVER | | |
| Source | **xs:IDREF** | required | | | |
| Dest | **xs:IDREF** | required | | | |
| Name | **xs:string** | optional | | | |
| Color | **xs:NMTOKEN** | optional | NONE | | |
| ID | **xs:ID** | required | | | |

source

```
<xs:complexType name="Edge.type">
 <xs:group ref="Edge.content"/>
 <xs:attributeGroup ref="Edge.attlist"/>
</xs:complexType>
```

## complexType **Event.type**

diagram




namespace http://cons.aptima.com/schemas/odl

type extension of **Node.type**

properties base Node.type

children **Annotation**

used by element **Event**

attributes

| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | **xs:token** | optional | | | |
| ID | **xs:ID** | required | | | |
| NodeProbability | **Probability** | optional | | | |
| BeginApplicability | **Date** | optional | ALWAYS | | |
| EndApplicability | **Date** | optional | NEVER | | |
| Goal | **xs:IDREF** | optional | | | |

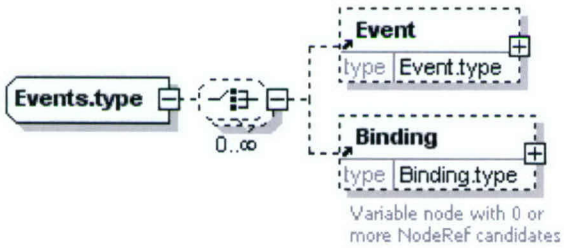
source

```
<xs:complexType name="Event.type">
```

```
  <xs:complexContent>
   <xs:extension base="Node.type">
    <xs:group ref="Event.content"/>
    <xs:attributeGroup ref="Event.attlist"/>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
```

complexType **Events.type**

diagram




namespace http://cons.aptima.com/schemas/odl

children **Event** **Binding**

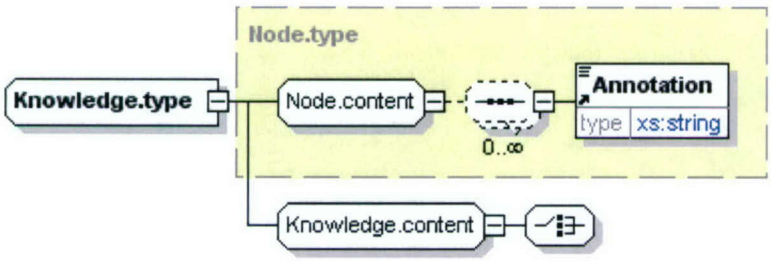
used by element **Events**

source

```
<xs:complexType name="Events.type">
 <xs:choice minOccurs="0" maxOccurs="unbounded">
  <xs:element ref="Event" minOccurs="0"/>
  <xs:element ref="Binding" minOccurs="0"/>
 </xs:choice>
</xs:complexType>
```

complexType **Knowledge.type**

diagram




namespace http://cons.aptima.com/schemas/odl

type extension of **Node.type**

properties base Node.type

children **Annotation**

used by element **Knowledge**

attributes

| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | **xs:token** | optional | | | |

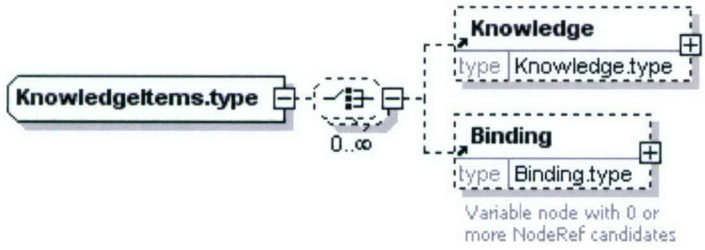
| | | | |
|---|---|---|---|
| ID | **xs:ID** | required | |
| NodeProbability | **Probability** | optional | |
| BeginApplicability | **Date** | optional | ALWAYS |
| EndApplicability | **Date** | optional | NEVER |
| Fact | **xs:string** | optional | |
| KnowledgeBase | **xs:IDREFS** | optional | |

source

```
<xs:complexType name="Knowledge.type">
 <xs:complexContent>
  <xs:extension base="Node.type">
   <xs:group ref="Knowledge.content"/>
   <xs:attributeGroup ref="Knowledge.attlist"/>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

complexType **KnowledgeItems.type**

diagram




namespace http://cons.aptima.com/schemas/odl

children **Knowledge** **Binding**

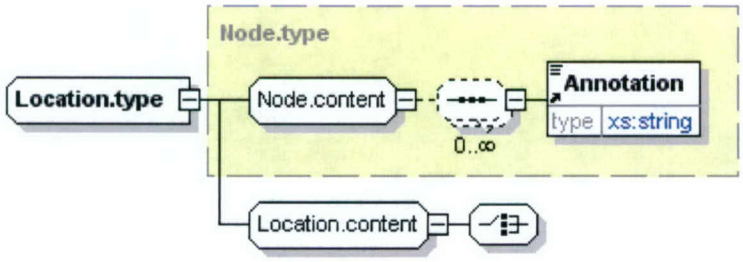
used by element **KnowledgeItems**

source

```
<xs:complexType name="KnowledgeItems.type">
 <xs:choice minOccurs="0" maxOccurs="unbounded">
  <xs:element ref="Knowledge" minOccurs="0"/>
  <xs:element ref="Binding" minOccurs="0"/>
 </xs:choice>
</xs:complexType>
```

complexType **Location.type**

diagram




namespace http://cons.aptima.com/schemas/odl

type extension of **Node.type**

properties base Node.type

children **Annotation**

used by element **Location**

attributes

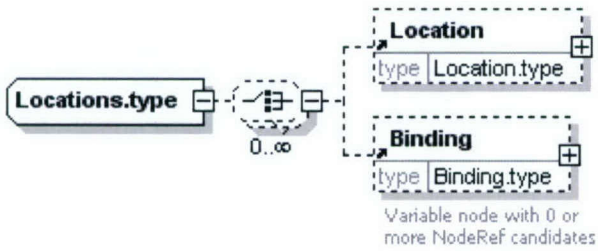
| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | xs:token | optional | | | |
| ID | xs:ID | required | | | |
| NodeProbability | Probability | optional | | | |
| BeginApplicability | Date | optional | ALWAYS | | |
| EndApplicability | Date | optional | NEVER | | |

source

```
<xs:complexType name="Location.type">
 <xs:complexContent>
  <xs:extension base="Node.type">
   <xs:group ref="Location.content"/>
   <xs:attributeGroup ref="Location.attlist"/>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

complexType **Locations.type**

diagram




namespace http://cons.aptima.com/schemas/odl

children **Location** **Binding**

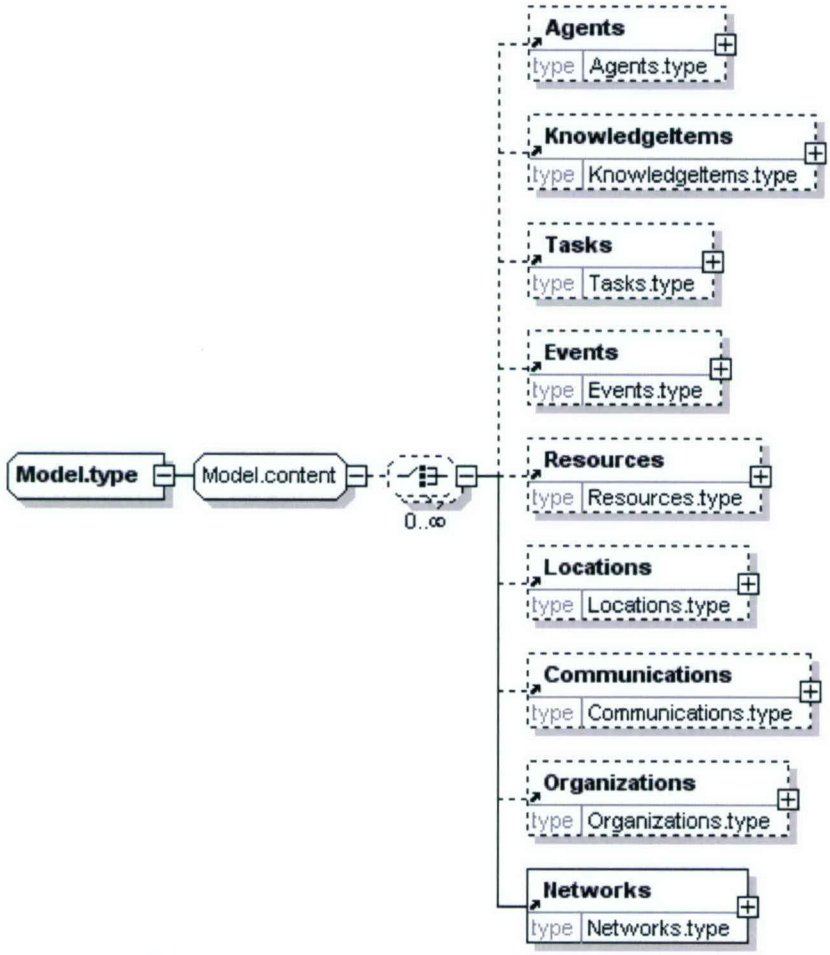
used by element **Locations**

source

```
<xs:complexType name="Locations.type">
 <xs:choice minOccurs="0" maxOccurs="unbounded">
  <xs:element ref="Location" minOccurs="0"/>
  <xs:element ref="Binding" minOccurs="0"/>
 </xs:choice>
</xs:complexType>
```

complexType **Model.type**

diagram




namespace http://cons.aptima.com/schemas/odl

children Agents KnowledgeItems Tasks Events Resources Locations Communications Organizations Networks

used by element Model

attributes

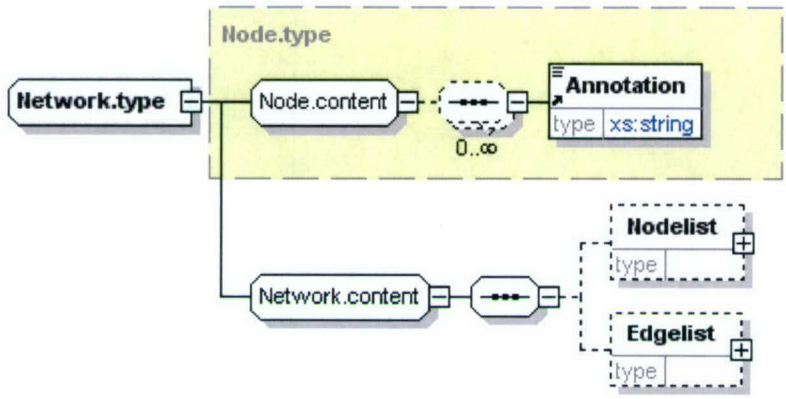
| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|

source

```
<xs:complexType name="Model.type">
  <xs:group ref="Model.content"/>
  <xs:attributeGroup ref="Model.attlist"/>
</xs:complexType>
```

complexType **Network.type**

diagram




namespace http://cons.aptima.com/schemas/odl

type extension of **Node.type**

properties base Node.type

children **Annotation** **Nodelist** **Edgelist**

used by element **Network**

attributes

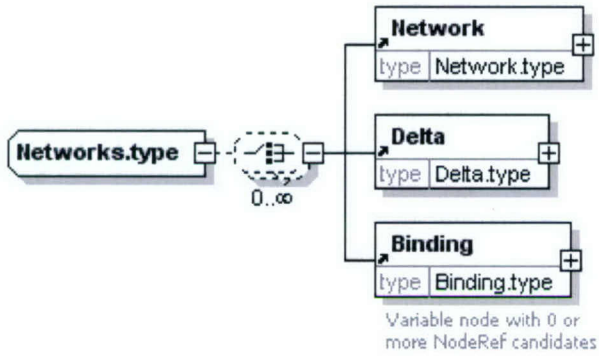
| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | **xs:token** | optional | | | |
| ID | **xs:ID** | required | | | |
| NodeProbability | **Probability** | optional | | | |
| BeginApplicability | **Date** | optional | ALWAYS | | |
| EndApplicability | **Date** | optional | NEVER | | |

source

```
<xs:complexType name="Network.type">
 <xs:complexContent>
  <xs:extension base="Node.type">
   <xs:group ref="Network.content"/>
   <xs:attributeGroup ref="Network.attlist"/>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

## complexType **Networks.type**

diagram




namespace http://cons.aptima.com/schemas/odl

children Network Delta Binding

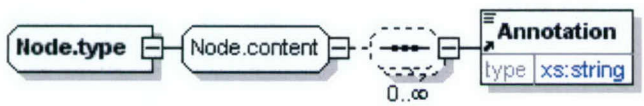
used by element Networks

source

```
<xs:complexType name="Networks.type">
 <xs:choice minOccurs="0" maxOccurs="unbounded">
  <xs:element ref="Network"/>
  <xs:element ref="Delta"/>
  <xs:element ref="Binding"/>
 </xs:choice>
</xs:complexType>
```

## complexType **Node.type**

diagram




namespace http://cons.aptima.com/schemas/odl

properties abstract true

children Annotation

used by element Node
complexTypes Agent.type Binding.type Communication.type Delta.type Event.type Knowledge.type Location.type Network.type Organization.type Resource.type Task.type

attributes

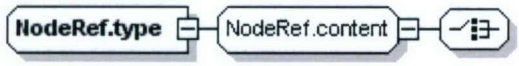
| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | xs:token | optional | | | |
| ID | xs:ID | required | | | |
| NodeProbability | Probability | optional | | | |
| BeginApplicability | Date | optional | ALWAYS | | |
| EndApplicability | Date | optional | NEVER | | |

source

```
<xs:complexType name="Node.type" abstract="true">
 <xs:group ref="Node.content"/>
 <xs:attributeGroup ref="Node.attlist"/>
</xs:complexType>
```

## complexType **NodeRef.type**

diagram




namespace http://cons.aptima.com/schemas/odl

used by element **NodeRef**

attributes

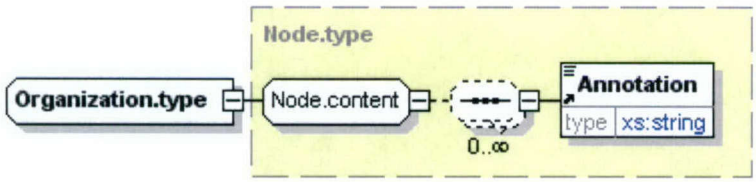
| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| ProbabilityOfNode | **Probability** | optional | 1 | | |
| Name | **xs:string** | optional | | | |
| Node | **xs:IDREF** | required | | | |

source

```
<xs:complexType name="NodeRef.type">
 <xs:group ref="NodeRef.content"/>
 <xs:attributeGroup ref="NodeRef.attlist"/>
</xs:complexType>
```

## complexType **Organization.type**

diagram




namespace http://cons.aptima.com/schemas/odl

type extension of **Node.type**

properties base Node.type

children **Annotation**

used by element **Organization**

attributes

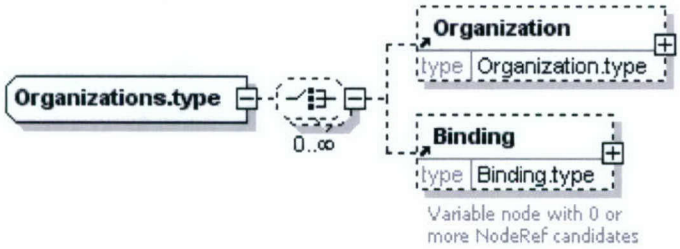
| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | **xs:token** | optional | | | |
| ID | **xs:ID** | required | | | |
| NodeProbability | **Probability** | optional | | | |
| BeginApplicability | **Date** | optional | ALWAYS | | |
| EndApplicability | **Date** | optional | NEVER | | |
| Network | **xs:IDREF** | optional | | | |

source

```
<xs:complexType name="Organization.type">
 <xs:complexContent>
  <xs:extension base="Node.type">
   <xs:attributeGroup ref="Organization.attlist"/>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

## complexType **Organizations.type**

diagram




namespace http://cons.aptima.com/schemas/odl

children Organization Binding

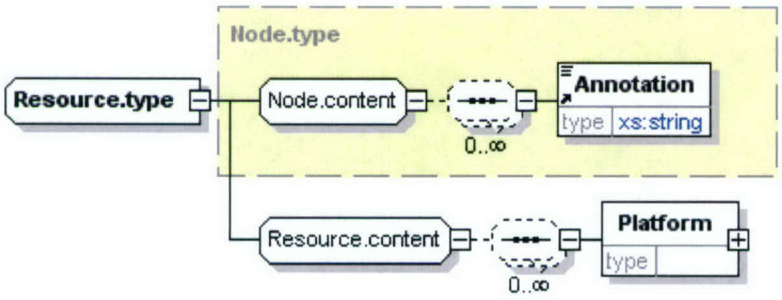
used by element Organizations

source

```
<xs:complexType name="Organizations.type">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element ref="Organization" minOccurs="0"/>
    <xs:element ref="Binding" minOccurs="0"/>
  </xs:choice>
</xs:complexType>
```

## complexType **Resource.type**

diagram




namespace http://cons.aptima.com/schemas/odl

type extension of Node.type

properties base Node.type

children Annotation Platform

used by element Resource

attributes

| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | xs:token | optional | | | |
| ID | xs:ID | required | | | |
| NodeProbability | Probability | optional | | | |
| BeginApplicability | Date | optional | ALWAYS | | |
| EndApplicability | Date | optional | NEVER | | |

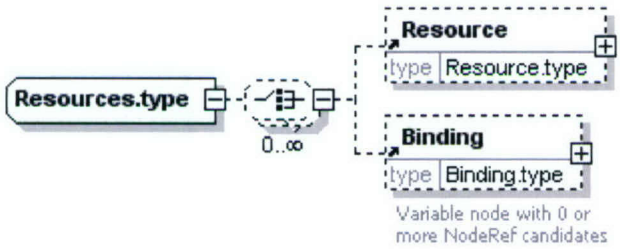
source

```
<xs:complexType name="Resource.type">
  <xs:complexContent>
```

```
    <xs:extension base="Node.type">
     <xs:group ref="Resource.content"/>
     <xs:attributeGroup ref="Resource.attlist"/>
    </xs:extension>
   </xs:complexContent>
  </xs:complexType>
```

complexType **Resources.type**

diagram




namespace http://cons.aptima.com/schemas/odl

children Resource Binding

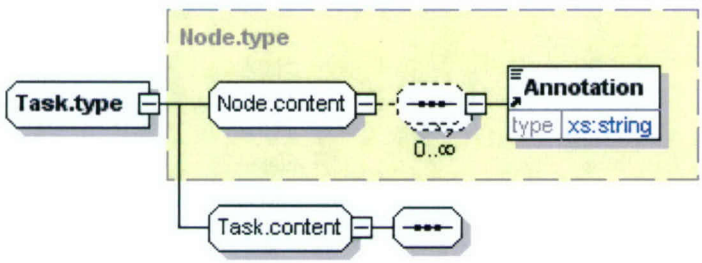
used by element Resources

source

```
<xs:complexType name="Resources.type">
 <xs:choice minOccurs="0" maxOccurs="unbounded">
  <xs:element ref="Resource" minOccurs="0"/>
  <xs:element ref="Binding" minOccurs="0"/>
 </xs:choice>
</xs:complexType>
```

complexType **Task.type**

diagram




namespace http://cons.aptima.com/schemas/odl

type extension of Node.type

properties base Node.type

children Annotation

used by element Task

attributes

| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | xs:token | optional | | | |
| ID | xs:ID | required | | | |

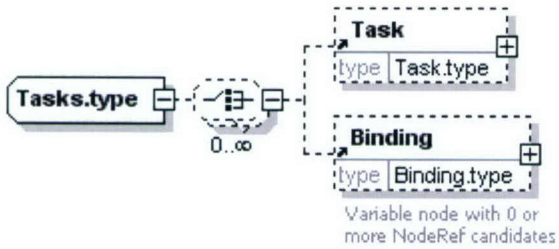
| | | | |
|---|---|---|---|
| NodeProbability | Probability | optional | |
| BeginApplicability | Date | optional | ALWAYS |
| EndApplicability | Date | optional | NEVER |
| PlannedStart | xs:dateTime | optional | |
| PlannedEnd | xs:dateTime | optional | |
| ActualStart | xs:dateTime | optional | |
| ActualEnd | xs:dateTime | optional | |
| Dependencies | xs:IDREFS | optional | |

source

```
<xs:complexType name="Task.type">
 <xs:complexContent>
  <xs:extension base="Node.type">
   <xs:group ref="Task.content"/>
   <xs:attributeGroup ref="Task.attlist"/>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

## complexType **Tasks.type**

diagram




namespace http://cons.aptima.com/schemas/odl

children Task Binding

used by element Tasks

source

```
<xs:complexType name="Tasks.type">
 <xs:choice minOccurs="0" maxOccurs="unbounded">
  <xs:element ref="Task" minOccurs="0"/>
  <xs:element ref="Binding" minOccurs="0"/>
 </xs:choice>
</xs:complexType>
```

## simpleType **Date**

namespace http://cons.aptima.com/schemas/odl

type union of (**xs:string**, PersistentDate)

used by attributes Node.attlist/@BeginApplicability Edge.attlist/@BeginApplicability Node.attlist/@EndApplicability Edge.attlist/@EndApplicability

source

```
<xs:simpleType name="Date">
 <xs:union memberTypes="xs:string PersistentDate"/>
</xs:simpleType>
```

simpleType **DeltaAction**

namespace http://cons.aptima.com/schemas/odl

type restriction of **xs:NMTOKEN**

used by attributes Delta.content/NodeDelta/@Action Delta.content/EdgeDelta/@Action

facets
enumeration ADD
enumeration DELETE
enumeration MODIFY

source
```
<xs:simpleType name="DeltaAction">
 <xs:restriction base="xs:NMTOKEN">
  <xs:enumeration value="ADD"/>
  <xs:enumeration value="DELETE"/>
  <xs:enumeration value="MODIFY"/>
 </xs:restriction>
</xs:simpleType>
```

simpleType **Mobility**

namespace http://cons.aptima.com/schemas/odl

type restriction of **xs:NMTOKEN**

used by attribute Resource.content/Platform/@Mobility

facets
enumeration FIXED
enumeration MOBILE

annotation
documentation MOBILE or FIXED

source
```
<xs:simpleType name="Mobility">
 <xs:annotation>
  <xs:documentation>MOBILE or FIXED</xs:documentation>
 </xs:annotation>
 <xs:restriction base="xs:NMTOKEN">
  <xs:enumeration value="FIXED"/>
  <xs:enumeration value="MOBILE"/>
 </xs:restriction>
</xs:simpleType>
```

simpleType **NodeTypeEnum**

namespace http://cons.aptima.com/schemas/odl

type restriction of **xs:NMTOKEN**

used by attribute Binding.attlist/@VariableClass

facets
enumeration AGENT
enumeration KNOWLEDGE
enumeration RESOURCE
enumeration TASK
enumeration EVENT
enumeration LOCATION
enumeration COMMUNICATION
enumeration ORGANIZATION

source
```
<xs:simpleType name="NodeTypeEnum">
 <xs:restriction base="xs:NMTOKEN">
  <xs:enumeration value="AGENT"/>
  <xs:enumeration value="KNOWLEDGE"/>
  <xs:enumeration value="RESOURCE"/>
  <xs:enumeration value="TASK"/>
  <xs:enumeration value="EVENT"/>
  <xs:enumeration value="LOCATION"/>
  <xs:enumeration value="COMMUNICATION"/>
  <xs:enumeration value="ORGANIZATION"/>
 </xs:restriction>
</xs:simpleType>
```

simpleType **PersistentDate**

namespace http://cons.aptima.com/schemas/odl

type restriction of **xs:NMTOKEN**

used by simpleType **Date**

facets enumeration ALWAYS
enumeration NEVER

source
```
<xs:simpleType name="PersistentDate">
 <xs:restriction base="xs:NMTOKEN">
  <xs:enumeration value="ALWAYS"/>
  <xs:enumeration value="NEVER"/>
 </xs:restriction>
</xs:simpleType>
```

simpleType **Probability**

namespace http://cons.aptima.com/schemas/odl

type restriction of **xs:double**

properties final restriction

used by attributes **Edge.attlist/@EdgeProbability** **Node.attlist/@NodeProbability** **NodeRef.attlist/@ProbabilityOfNode** **Edge.attlist/@TransitionProbability**

facets minInclusive 0
maxInclusive 1

annotation documentation Double between 0 and 1 inclusive

source
```
<xs:simpleType name="Probability" final="restriction">
 <xs:annotation>
  <xs:documentation>Double between 0 and 1 inclusive</xs:documentation>
 </xs:annotation>
 <xs:restriction base="xs:double">
  <xs:minInclusive value="0"/>
  <xs:maxInclusive value="1"/>
 </xs:restriction>
```

```
</xs:simpleType>
```

attributeGroup **Agent.attlist**

namespace http://cons.aptima.com/schemas/odl

used by complexType **Agent.type**

source

```
<xs:attributeGroup name="Agent.attlist"/>
```

attributeGroup **Binding.attlist**

namespace http://cons.aptima.com/schemas/odl

used by complexType **Binding.type**

attributes

| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| VariableClass | NodeTypeEnum | optional | | | |

source

```
<xs:attributeGroup name="Binding.attlist">
 <xs:attribute name="VariableClass" type="NodeTypeEnum" use="optional"/>
</xs:attributeGroup>
```

attributeGroup **Communication.attlist**

namespace http://cons.aptima.com/schemas/odl

used by complexType **Communication.type**

attributes

| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| From | xs:IDREF | optional | | | |
| To | xs:IDREF | optional | | | |

source

```
<xs:attributeGroup name="Communication.attlist">
 <xs:attribute name="From" type="xs:IDREF" use="optional"/>
 <xs:attribute name="To" type="xs:IDREF" use="optional"/>
</xs:attributeGroup>
```

attributeGroup **Delta.attlist**

namespace http://cons.aptima.com/schemas/odl

used by complexType **Delta.type**

attributes

| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| AppliesTo | xs:IDREF | required | | | |

source

```
<xs:attributeGroup name="Delta.attlist">
 <xs:attribute name="AppliesTo" type="xs:IDREF" use="required"/>
</xs:attributeGroup>
```

attributeGroup **Edge.attlist**

namespace http://cons.aptima.com/schemas/odl

used by complexType **Edge.type**

attributes

| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|

| | | | |
|---|---|---|---|
| EdgeProbability | Probability | optional | |
| TransitionProbability | Probability | optional | |
| BeginApplicability | Date | optional | ALWAYS |
| EndApplicability | Date | optional | NEVER |
| Source | xs:IDREF | required | |
| Dest | xs:IDREF | required | |
| Name | xs:string | optional | |
| Color | xs:NMTOKEN | optional | NONE |
| ID | xs:ID | required | |

source

```
<xs:attributeGroup name="Edge.attlist">
 <xs:attribute name="EdgeProbability" type="Probability" use="optional"/>
 <xs:attribute name="TransitionProbability" type="Probability" use="optional"/>
 <xs:attribute name="BeginApplicability" type="Date" use="optional"
default="ALWAYS"/>
 <xs:attribute name="EndApplicability" type="Date" use="optional"
default="NEVER"/>
 <xs:attribute name="Source" type="xs:IDREF" use="required"/>
 <xs:attribute name="Dest" type="xs:IDREF" use="required"/>
 <xs:attribute name="Name" type="xs:string" use="optional"/>
 <xs:attribute name="Color" type="xs:NMTOKEN" use="optional"
default="NONE"/>
 <xs:attribute name="ID" type="xs:ID" use="required"/>
</xs:attributeGroup>
```

attributeGroup **Event.attlist**

namespace http://cons.aptima.com/schemas/odl

used by complexType **Event.type**

attributes

| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Goal | xs:IDREF | optional | | | |

source

```
<xs:attributeGroup name="Event.attlist">
 <xs:attribute name="Goal" type="xs:IDREF" use="optional"/>
</xs:attributeGroup>
```

attributeGroup **Knowledge.attlist**

namespace http://cons.aptima.com/schemas/odl

used by complexType **Knowledge.type**

attributes

| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Fact | xs:string | optional | | | |
| KnowledgeBase | xs:IDREFS | optional | | | |

source

```
<xs:attributeGroup name="Knowledge.attlist">
 <xs:attribute name="Fact" type="xs:string" use="optional"/>
 <xs:attribute name="KnowledgeBase" type="xs:IDREFS" use="optional"/>
</xs:attributeGroup>
```

attributeGroup **Location.attlist**

namespace http://cons.aptima.com/schemas/odl

used by complexType **Location.type**

source
```
<xs:attributeGroup name="Location.attlist"/>
```

attributeGroup **Model.attlist**

namespace http://cons.aptima.com/schemas/odl

used by complexType **Model.type**

source
```
<xs:attributeGroup name="Model.attlist"/>
```

attributeGroup **Network.attlist**

namespace http://cons.aptima.com/schemas/odl

used by complexType **Network.type**

source
```
<xs:attributeGroup name="Network.attlist"/>
```

attributeGroup **Node.attlist**

namespace http://cons.aptima.com/schemas/odl

used by complexType **Node.type**

attributes

| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Name | **xs:token** | optional | | | |
| ID | **xs:ID** | required | | | |
| NodeProbability | **Probability** | optional | | | |
| BeginApplicability | **Date** | optional | ALWAYS | | |
| EndApplicability | **Date** | optional | NEVER | | |

source
```
<xs:attributeGroup name="Node.attlist">
 <xs:attribute name="Name" type="xs:token" use="optional"/>
 <xs:attribute name="ID" type="xs:ID" use="required"/>
 <xs:attribute name="NodeProbability" type="Probability" use="optional"/>
 <xs:attribute name="BeginApplicability" type="Date" use="optional"
default="ALWAYS"/>
 <xs:attribute name="EndApplicability" type="Date" use="optional"
default="NEVER"/>
</xs:attributeGroup>
```

attributeGroup **NodeRef.attlist**

namespace http://cons.aptima.com/schemas/odl

used by complexType **NodeRef.type**

attributes

| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| ProbabilityOfNode | **Probability** | optional | 1 | | |

| | | |
|---|---|---|
| Name | **xs:string** | optional |
| Node | **xs:IDREF** | required |

source
```
<xs:attributeGroup name="NodeRef.attlist">
 <xs:attribute name="ProbabilityOfNode" type="Probability" use="optional"
default="1"/>
 <xs:attribute name="Name" type="xs:string" use="optional"/>
 <xs:attribute name="Node" type="xs:IDREF" use="required"/>
</xs:attributeGroup>
```

attributeGroup **Organization.attlist**

namespace http://cons.aptima.com/schemas/odl

used by complexType **Organization.type**

attributes

| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| Network | **xs:IDREF** | optional | | | |

source
```
<xs:attributeGroup name="Organization.attlist">
 <xs:attribute name="Network" type="xs:IDREF" use="optional"/>
</xs:attributeGroup>
```

attributeGroup **Resource.attlist**

namespace http://cons.aptima.com/schemas/odl

used by complexType **Resource.type**

source
```
<xs:attributeGroup name="Resource.attlist"/>
```

attributeGroup **Task.attlist**

namespace http://cons.aptima.com/schemas/odl

used by complexType **Task.type**

attributes

| Name | Type | Use | Default | Fixed | Annotation |
|---|---|---|---|---|---|
| PlannedStart | **xs:dateTime** | optional | | | |
| PlannedEnd | **xs:dateTime** | optional | | | |
| ActualStart | **xs:dateTime** | optional | | | |
| ActualEnd | **xs:dateTime** | optional | | | |
| Dependencies | **xs:IDREFS** | optional | | | |

source
```
<xs:attributeGroup name="Task.attlist">
 <xs:attribute name="PlannedStart" type="xs:dateTime" use="optional"/>
 <xs:attribute name="PlannedEnd" type="xs:dateTime" use="optional"/>
 <xs:attribute name="ActualStart" type="xs:dateTime" use="optional"/>
 <xs:attribute name="ActualEnd" type="xs:dateTime" use="optional"/>
 <xs:attribute name="Dependencies" type="xs:IDREFS" use="optional"/>
</xs:attributeGroup>
```